Combining Machine Learning with Computer Vision for Precision Agriculture Applications

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Dimitris Zermas

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Nikolaos Papanikolopoulos

April, 2018

Acknowledgements

This thesis is a testament of the shaping process that turned me into the person I am today. This shaping would not be possible without the people that acted as grinding stones and nourishing elements throughout my life. I am grateful towards my parents Yiorgos and Efrosyni for their efforts to bring up a decent human raised with love, and towards my sister Xanthippe who taught me that this love was to be shared.

I would like to express my sincere gratitude to my advisor Prof. Papanikolopoulos for the continuous support of my Ph.D study, for his patience, and motivation. His trust in my capabilities allowed me to reveal and refine them. I could not have wished for a more supportive mentor. I also want to thank my co-advisor Prof. Mulla for his willingness to share his knowledge and genuine excitement for my research which gave me the strength to move forward after each stall.

My sincere thanks goes to Dr. Morellas for the time and care he devoted to me and my doubtful thoughts. Prof. Meyer, as a member of my committee, was present from the start with his insightful comments and I thank him for that. From the team that worked hard to realize this thesis I want to thank Mr. Bazakos for his tenacity to initiate such a project and Prof. Kaiser for his valuable assistance in collecting the data.

I thank my colleagues and friends Panos, Bill, Duc, Dario, Ruben, Josh, Nick, Tyler, Aicam, Dimitris, and Yiannis for their stimulating discussions, and the community around HSA for all the fun we had in the last years. I also thank my friends and mentors at the U of Patras, this is where it all started!

This section would not be complete without acknowledging those who trusted my capabilities in very stressful times and these are the people at DELPHI in Agoura Hills, CA and at Sentera in Minneapolis, MN.

Last but certainly not least I want to thank my wife, Fotini, who appeared in my life to provide the stability, care, and genuine love I needed to conclude my academic efforts.

Dedication

To my supporting pillars; my parents and Fotini

Abstract

Financial and social elements of modern societies are closely connected to the cultivation of corn. Due to its massive production, deficiencies during the cultivation process directly translate to major financial losses. Existing field monitoring solutions utilize aerial and ground means towards identifying sectors of the farmland presenting under-performing crops. Nevertheless, an inference element is still absent; that is the automated diagnose of the cause and severity of the deficiency. The early detection and treatment of crops deficiencies and the frequent evaluation of their growth status are thus tasks of great significance. Towards an automated health condition assessment, this thesis introduces schemes for the computation of plant health indices.

First, we propose a methodology to detect nitrogen (N) deficiencies in corn fields and assess their severity at an early stage using low-cost RGB sensors. The introduced methodology is twofold. First, a low complexity recommendation scheme identifies candidate plants exhibiting nitrogen deficiency and second, a detection elimination step completes the inference loop by deciding which of the candidate plants are actually exhibiting that condition. Experimental results on a diverse real-world dataset achieve a 90.6% accuracy for the detection of N-deficient regions and support the extension of this methodology to other crops and deficiencies that show similar visual characteristics.

Second, based on the 3D reconstruction of small batches of corn plants at growth stages between "V3" and "V6", an automated alternative to existing manual and cumbersome phenotype estimation methodologies is presented. The use of 3D models provides an elevated information content, when compared to planar methods, mainly due to the alleviation of leaf occlusions. High-resolution images of corn stalks are collected and used to obtain 3D models of plants of interest. Based on the extracted 3D point clouds, the calculation of a plethora of phenotypic characteristics for each 3D reconstruction are obtained such as the number of plants depicted with 88.1% accuracy, Leaf Area Index (LAI) with 92.48% accuracy, the height with 89.2% accuracy, the leaf length with 74.8% accuracy, and the location and the angles of leaves with respect to the stem. The last two variables are connected by showing the trend of the angles to change with respect to the leaf position on the stem as the crops grow. An experimental validation

using both artificially made corn plants emulating real-world scenarios and real corn plants in different growth stages supports the efficacy of the proposed methodology.

Although the proposed methodologies are agnostic to the platform that performs the data collection, for the presented experiments a MikroKopter Okto XL equipped with a Nikon D7200 RGB sensor and a DJI Matrice 100 with a Zenmuse X3 and a Zenmuze Z3 RGB high-resolution cameras were used. The flight altitude ranged between 6 and 15 m and the resolution of the images varies within a range of 0.2 to 0.47 cm/pixel.

Thorough data collection and interpretation leads to a better understanding of the needs not only of the farm as a whole but to each individual plant providing a much higher granularity to potential treatment strategies. Through the thoughtful utilization of modern computer vision techniques, it is possible to achieve positive financial and environmental results for these tasks. The conclusions of this work, suggest a fully automated scheme for information gathering in modern farms capable of replacing current labor-intensive procedures, thus greatly impacting the timely detection of crop deficiencies.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The cultivation of corn has a substantial financial impact in the global economy. According to the U.S. National Corn Growers Association (NCGA), 41.1 billion bushels of corn were harvested worldwide for the year of 2017, with an estimated gross value of $133.6 billion [1]. The financial impact of corn can be further understood if one studies the particular production challenges in a localized manner with the example of the sate of Minnesota where there are 3.4 million hectares of corn cropland. For a state of 5.5 million residents, these generate over $4.1 billion in revenue and employ nearly 1 in 5 of Minnesota's workers.

Modern corn production farms average over 1,100 acres according to statistics from the U.S. Department of Agriculture (USDA) [2] making the surveillance and tracking of crops growth at such a large scale a non-trivial task. Furthermore, the spatial and temporal variability of the deficiencies makes diagnosis and treatment difficult and leads to crop losses. As an example, nutrient deficiencies such as nitrogen (N) can limit the productivity of corn crops up to 20% [3].

Currently, crop stresses are primarily identified using crop scouting by trained professionals. These professionals cannot efficiently cover all crop acreage, as this process is time consuming. As a result, fertilizer is applied at uniform rates on farms, without regard to the spatial or temporal variability in crop nutrient deficiencies. This inefficiency leads to excess fertilizer application, which ends up contaminating ground and underground water sources. As an example, Minnesota cropland loses on average 114 million kg of nitrate that pollutes the Mississippi River, contributing to hypoxia in the

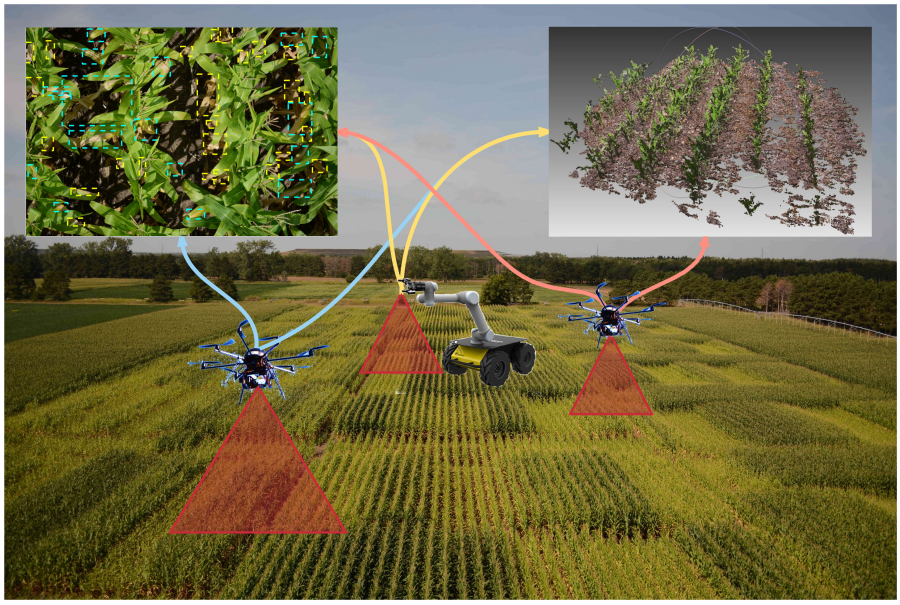


Figure 1.1: Envisioned autonomous robotic inspection of crop fields for the detection, mapping and characterization of nutrient deficiencies and the collection of biometrics. Any robotic platform would employ low–cost sensors to acquire data in multiple spatial and temporal resolutions in order to provide comprehensive results, early detection, and automated fertilizer recommendation. Two types of aerial and ground platforms are presented in the photo with results of the Ndeficiency and plant 3D reconstruction on the upper left and right corners respectively.

Gulf of Mexico.

An alternative to uniform applications of fertilizer is precision applications that take into account the right location and time. Precision Agriculture (PA) is a scientific domain that offers to improve crop productivity and farm profitability through improved management of farm inputs, leading to better environmental quality. Unfortunately, PA has not yet produced a comprehensive answer for the early identification of large-scale field deficiencies. Current approaches rely on either laborious, manual data collection or remote sensing methodologies that utilize very low resolution image data from satellites, or airplanes. Satellite and aerial remote sensing is severely limited by cloud cover and deficiencies might not be diagnosed early enough to prevent crop losses.

Recently, in PA alternative approaches have introduced robotics through the use of Unmanned Aerial Vehicles (UAVs) and these initiatives are expected to financially impact the agriculture domain. In Minnesota alone, the Dept. of Employment and

Economic Development Office estimates that UAVs alone in precision agriculture will generate $150 million in new revenue and create a thousand new jobs in the next two years.

This thesis brings together PA, Computer Vision (CV), and Machine Learning (ML) for an automated plant pathology assessment tailored around corn production. The aim is to develop an integrative framework which will provide a comprehensive pallet of information to facilitate the maximization of corn yield and minimize the application of fertilizers. The thesis addresses two complementary problems of major importance for yield maximization;

- the Nstress detection in corn which facilitates the fertilization process through detailed information on the location and severity of the stress, and
- the estimation of several corn phenotypic characteristics through an original interpretation of 3D models produced by collections of 2D images.

As shown in Figure 1.1, we envision that any robotic platform with the ability to provide high resolution top-down imagery or video will be able to utilize the developed algorithms and extract data that help in the early identification of deficiencies through the developed image processing, 3D reconstruction and point cloud processing algorithms. The proposed framework can revolutionize the field and benefit agriculture in the U.S. and the world.

Our experiments conclude that Ndeficiency can be detected through high resolution aerial imaging when the corn plants show visible symptoms with 90.6% accuracy, while there are signs that the same methodology can be transferred to other crops and deficiencies. Further, we show that phenotypic characteristics of individual plants can be extracted automatically with high accuracy based on a 3D model. Our results include the individual plant segmentation and counting from a given 3D reconstructed field scene with 88.1% accuracy, the Leaf Area Index estimation with 92.48% accuracy, the individual plant height computation with 89.2% accuracy, the leaf length extraction with 74.8% accuracy, the measurement of angles between leaves and stems, and the distance between leaves of the same plant. We connect the last two variables showing the trend of the angles to change with respect to the leaf position on the stem as the crops grow.

Chapter 2

Literature Review

This chapter provides a brief introduction to the corn life cycle, as well as an overview of research activities in field nutrient deficiency detection, and crops 3D reconstruction attempts.

2.1 Corn Plant Growth Stages

Corn follows systematic stages of growth that depend on the number of leaves appearing on the stalk and the extent of grain filling [4]. During the initial development and before the tassels start becoming visible, the stages are characterized by the letter "V" followed by an integer corresponding to the number of collars developed by the plant. The "V" stages range from V1 to V12 with VT signifying the stage when the tassel emerges. V1 usually occurs in early June and V12 occurs in middle-late July. Once the tassels start becoming visible and till the harvesting stage, the letter "R" represents the reproductive stages when the kernel develops and matures. The reproductive stages range from R1 to R6 as shown in Fig. 2.1.

The Nuptake of the corn plant is restricted to the first ("V") part of its development which in turn necessitates the fertilization of the land by the V10 growth stage. Although this behaviour poses temporal constraints in the deployment of our system, at the same time it reduces the complexity of the scene to be analyzed by our algorithms and traversed by our robots. Essentially, we are looking into plants that are generally smaller than 4 feet height and have less than 12 leaves. This is a critical factor when

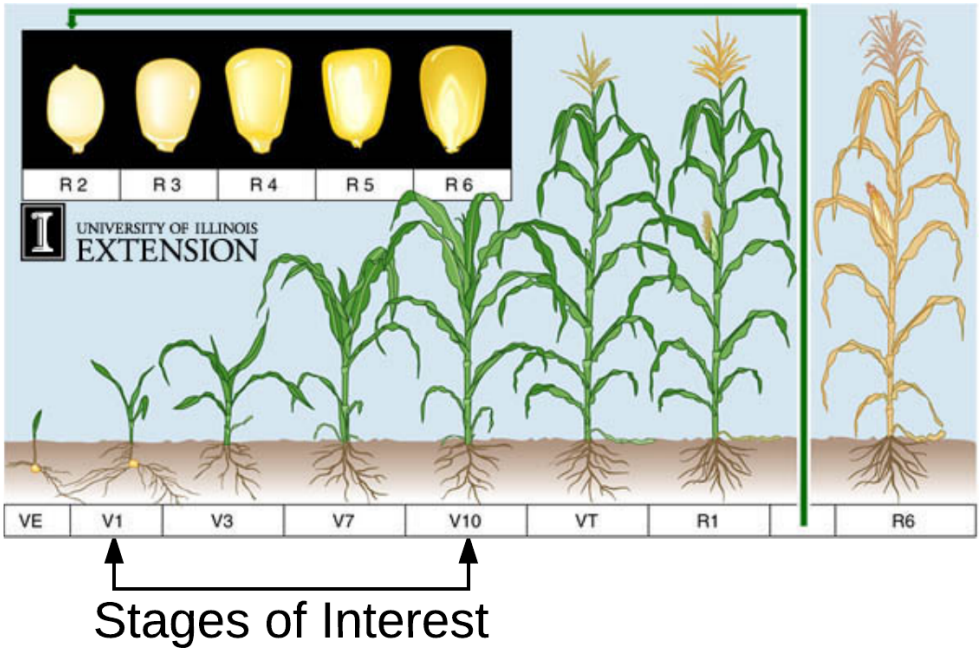


Figure 2.1: The growth stages of corn plants are characterized by a letter followed by a number (e.g. "V5"). The letter "V" symbolizes the vegetative stage of the corn, while the number shows how many leaves have grown a visible collar around the stem [4]. In particular, the *N*deficiencies start becoming visible at V5 and the plant is susceptible to treatment up to V10. This is the range (V5-V10) we are targeting to assess the *N*deficiency severity. Stage VT comes when the last branch of the tassel is completely visible, and the stages characterized by "R" indicate the reproductive stages when the kernel develops and matures. The image presented here was obtained from University of Illinois Extension.

designing our methods and considering their feasibility in real world situations.

2.2 Detection of Nutrient Deficiencies in Corn Fields

In an effort to meet the nutritious needs of the increasing world population, technological advances are incorporated into different sections of agriculture worldwide ranging from finance [5] to engineering [6] and PA. PA, as a research domain, is concerned with the yield maximization of farm fields by applying the right farm inputs at the right place and time. Advancements in the area of Remote Sensing (RS), which exploits sensors to gather data from remote locations, have been successfully integrated with PA to

assist agricultural applications. Examples of modern hand-held sensing technology for the extraction of high precision plant pathology information were performed by Des et al. [7] and Gealy et al. [8]. Mulla [9] presented an overview of the progress in PA over the last twenty five years and showed that technological advances in RS allowing non-contact field surveillance have greatly benefited it.

Regarding the visible spectrum, which is the focus of the proposed methodology, the most substantial attempts to classify diseases in crops and provide an estimate on their severity have been collected in two review papers by Barbedo [10, 11]. The author's discoveries reveal that many methodologies assume the background of the leaf images to be of a single color and the illumination conditions to be controlled. These observations shade light in the shortcomings that need to be addressed in order to construct a generic, automated, and reliable system for crop stress detection and classification.

For the detection of leaf diseases in real world fields under non-limiting image capturing conditions, segmentation based on color and texture plays a significant role. In their work, Wang et al. [12] utilize edge detection, skeletonization, and histogram thresholding to identify leaves of interest, but their method is limited since it is only applied on images that have been manually selected to depict a whole leaf. This attempt reveals that texture and gray-scale intensity alone are not suitable for a reliable and generic solution.

On the other hand, Guo et al. [13] utilize only color information in six different color spaces to train a random forest classifier for the separation of green from the ground. Their methodology requires a feature selection step that picks the most prominent color channels, and a training step that, according to the authors' claim, is performed only once and is capable of accommodating different illumination conditions. It is not clear if this approach would work under different soil types.

Similarly, Bai et al. [14] utilize only the CIE L*a*b*[15] colorspace to accomplish the segmentation of green leaves of rice against a water background with heavy reflections. This approach slices the L*a*b* space in planes perpendicular to the L* axis, and assumes that the color intensity invariance between consecutive planes is an indication of persistent green pixels. As there will be seen later in this work, utilizing only the L*a*b* colorspace performs poorly in cases with great variation in the green intensity. This phenomenon is dominant in the case of corn as opposed to rice, due to its larger

leaf area.

Several color segmentation methodologies with focus on the various colorspaces have been proposed by precision agriculture groups over the recent years [16, 17, 18, 19, 14, 13, 20]. In reality, the simple fact that different authors use different color space combinations to claim similar segmentation accuracy should be enough to raise questions on the value behind the pursue for the absolute best color channel combination.

One should keep in mind the seminal work of Gevers and Smeulders [21] who claim the significant correlation between most of the numerous colorspace channels. Specifically, they consider only a few channels to be *essentially different*; intensity I, RGB, normalized color rgb, and for the applications that involve a human in the loop; hue H and saturation S. From a machine learning point of view, the concatenation of correlated and linearly dependent spaces does not introduce new information and only hinders the algorithmic complexity by augmenting the solution space.

Another research branch that focuses on the leaf segmentation problem utilizes both texture and color information through the application of superpixels. The dominant superpixel algorithm for those approaches is the SLIC [22] which is essentially a K-means clustering in a 5D space (L*,a*,b* three pixel color values plus x,y two pixel coordinate values) with a modified distance objective function that groups neighboring pixels creating areas with homogeneous coloration. The resulting superpixels are assigned a class based on their color and several authors use different classifiers to achieve this. Examples are the works of Lu et al. [23] who employ a nearest neighbors classifier, Ye et al. [24] who consider neighboring superpixels and Markov random fields to assign a label, and Afridi et al. [25] who utilize unsupervised K-means on top of the superpixels to segment areas of interest.

A second line of research focuses on the classification of plant deficiencies where the training queries are handpicked [26], or generated by multiple sliding or randomly placed windows [27, 28, 29] on the images of scanned leaves. The query images undergo processing for the extraction of color, texture, and off-the-shelf features in an attempt to train the most accurate classifier targeted to a specific disease of a plant type.

In particular, Camargo and Smith [26] created and compared an extensive list of features regarding shape, color and texture hoping to create a generic disease classification process. In contrast, Pires et al. [29] consider some readily available image features

such as SIFT, SURF, and PHOW and discard low level color information. The same is true for Romualdo et al. [27] where the authors employ Gabor Wavelets to capture the texture of interest on N deficient corn plants.

Based on the available literature, putting together a feature space that will perform sufficiently under all deficiencies and plant types seems impossible. Nevertheless, it might be feasible to lay the foundation for easily adapting a robust pipeline to the needs of each application.

2.3 3D Reconstruction and Modelling of Plants

Although the need for a detailed 3D model of the corn canopy was apparent even in the early stages of PA [30, 31, 32], the 3D reconstruction applications in PA are limited and target mainly the estimation of plant biometrics from a high altitude and low granularity [33], whereas attempts for detailed 3D reconstruction and leaf area estimation are usually performed on a single plant in laboratories and under controlled conditions [34, 35, 36, 37, 38].

The first 3D reconstruction attempt in real world conditions targeting the leaf area estimation using Structure from Motion (SfM) was by Ivanov et al. [39], who utilized a stereo rig and manually segmented the leaves in the image pairs.

Since then, rarely researchers have tried to solve the problem of creating and utilizing a 3D model of plants in a densely populated field. Most relative is the work of Sodhi et al. [40] who are using a ground robotic platform to gather images of sorghum, perform 3D reconstruction with SfM, try to segment the stem from the leaves, and compute some biometrics. The experiments were completed in a complex environment, but the analysis was done on individual plants that seem to have been manually separated from the cluttered background.

Two related projects that can handle very small and well separated plants are presented in the works of Jin et al. [41] and Jay et al. [42]. In both these cases, the platforms that acquired the imagery were ground robots with cameras gathering data from a close distance to the plants.

Other notable publications regarding 3D reconstruction and measurements performed on plants in the field include Biskup et al. [43] that use stereo vision to get

geometric characteristics of the canopy, and Dong et al.[44] and Carlone et al. [45] who propose a model for the estimation of the plant's height over time. Recently, Qu et al.[46] presented a real time 3D reconstruction sensor for various agricultural applications, and Kjaer et al. [47] have been experimenting with 3D reconstruction from near-infrared 3D scanners in order to assess the nutrient state of plants in controlled environments.

On the side of mathematically modelling corn plants, an impressive 3D modelling for the purpose of studying plant reflectance was introduced by España et al. [48], who utilizes simple geometric primitives to capture the topology of the stem and leaves of corn. The same author utilizes these models to look into the bidirectional reflectance distribution function (BRDF) of corn canopies [49], an important characteristic that explains how much of the sun light is absorbed by the plant and is correlated with the plant health. More recently, the team of Fournier and Pradal [50] show a highly flexible and adaptive parametrized model for corn leaves which is accompanied by an algorithmic module that interacts with the open software platform OpenAlea [51].

Lastly, as summarized by Li et al. [52] and Kazmi et al. [53], several authors have considered solving the problem of acquiring detailed 3D models of individual plants using various sensors such as Light Detection And Ranging (LiDAR) [54, 55, 56] and fluorescent emitters [57]. These seems to be viable solutions with usual downsides being the sensor cost and size, and the very specific conditions for reliable data gathering.

Chapter 3

Detection of Nitrogen Deficiencies in Corn Fields

The surveillance and tracking of crops growth at a large scale is a non-trivial task. Solutions that rely on the monitoring of crops by humans and satellite images are time consuming, inefficient, and expensive. In contrast, Unmanned Aerial Vehicle (UAVs) solutions increase the efficiency of such tasks via the deployment of both fixed-wing and rotary platforms which can capture large collections of images in an automated manner. A plethora of sensors including RGB and hyper-spectral cameras have been integrated with UAVs to capture targeted information about the health condition of the plants. The collected images are then enriched with statistics associated with the state of the plants capitalizing on the reflectivity they exhibit in selected Near InfraRed (NIR) spectra [58]. Such monitoring schemes decrease the overall cost of the process while increasing the frequency that these tasks can be completed and enjoy a very large commercial success. The use of robotic automation that requires low cost platforms and sensors can economically justify their deployment in rural areas, addressing some of the valid concerns raised by Bechar et al. [59] regarding the role of agricultural robotics in a modern farm.

Even though locating stressed areas can be accomplished, an inference element of major importance to the farmers is lacking; that is the automated identification of the type and severity of the stress. This area of research has attracted significant

attention recently, with several precision agriculture groups attempting to detect and classify deficiencies of large volume cultivation plants. In his paper review, Barbedo [10] concludes that the existing methods share three main drawbacks; (i) lack of generality (methods are tuned for specific plant varieties and species), (ii) difficulty to apply in real world conditions, (iii) lack of tool sophistication that limits their applicability. The same author in [11] specifies the following two challenges that directly connect to the three main drawbacks and have not yet been satisfactorily addressed:

- The segmentation of regions of interest (ROI) in a busy image, and
- The processing of images with methods invariant to the capturing conditions.

The automation of this task has been neglected as it poses significant technical challenges and requires a combination of agriculture and algorithmic expertise. In this work we:

- propose a methodology that addresses these issues and test it extensively on detecting Nitrogen (N) deficiencies in corn plants, and
- show the feasibility of transferring this approach to other crops and deficiencies.

The lack of $N$is one of the most common nutrient deficiencies in corn is associated with the loss of 20% of the yearly yield [3]. This statement explains why many studies [60, 61] concentrate in Ndeficiency and treatment. Farmers, in an effort to reduce their yield loss, apply excessive amounts of fertilizer most of which ends up polluting the local water sources. Environmental studies on the repercussions of excess application of Nfertilizers in the USA farmlands, have shown that through the Mississippi river, a great amount of $N$is carried to the Gulf of Mexico creating a hypoxic zone - an area of low to no oxygen that can kill fish and marine life, also known as *the dead zone* [62].

Our experiments have shown a clear correlation of the number of N-deficient leaves with the fertilizer that has been applied before the seeding of the plants 3.2. This result reveals the need of an accurate method to estimate the number of deficient leaves. Our goal is the algorithmic instantiation of agronomists' expertise towards diagnosing Ndeficiencies in corn plants and assessing their severity in order to elevate the information conveyed to the farmer. Utilizing the latest advances in computer vision

Figure 3.1: The results of the classification scheme (Sec. 3.3). All the bounding boxes result from the recommendation scheme (Sec. 3.2). The yellow boxes are the ones that were classified as false suggestions while the bounded boxes in red dashed lines are those suggestions that have been classified as containing N-deficient leaves.

and machine learning, we address this problem in a real world environment by taking advantage of the unique signs that N-deficient plants exhibit. Nitrogen deficiency in corn appears as a "V"-shaped yellowing of leaf color, starting with tips of lower leaves (Fig. 3.3). We have proposed a scheme that utilizes the flying capabilities of UAVs to capture high resolution images of small groups of corn plants, while flying at low altitude, and consequently pinpoint the leaves that exhibit Ndeficiency.

The application of this methodology in real field data achieves an 90.6% accuracy in determining that a suggested area indeed has a nitrogen deficiency. In Fig. 3.1 the results of our classification scheme provide encouragement for the impact the advancement of this technology can make.

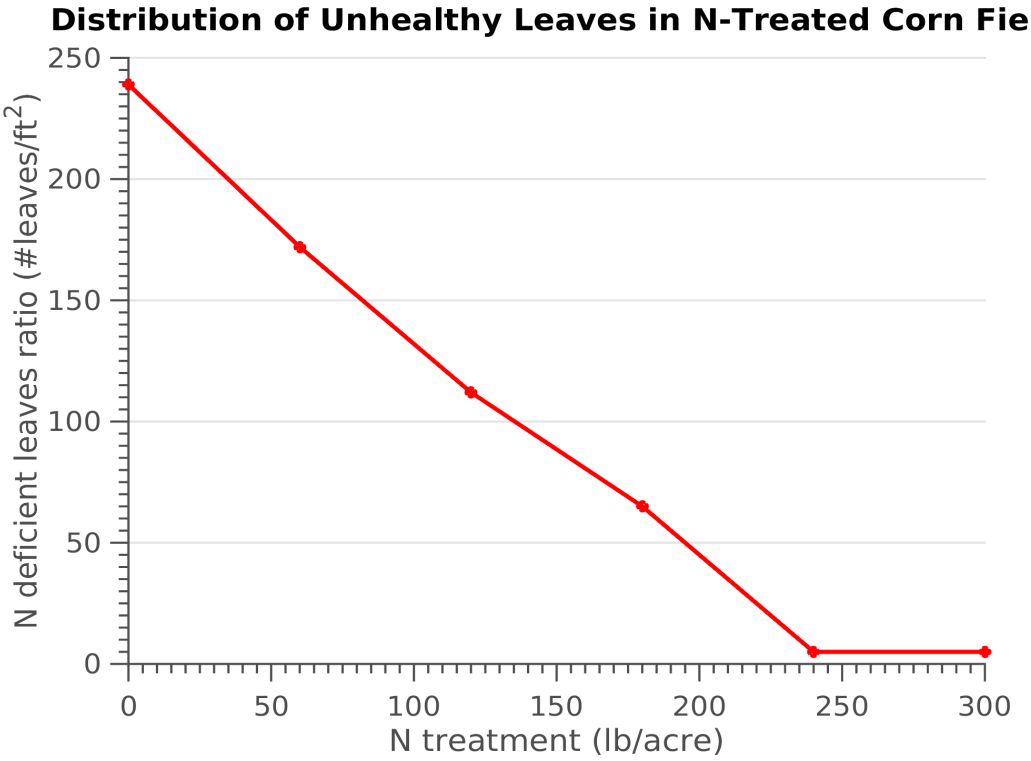


Figure 3.2: The correspondence between the number of verified Ndeficient leaves at V12 growth stage and the amount of fertilizer applied at the beginning of the cultivation period seems to be linear. A plot in this study is defined as a boxed area of 420 plants distributed over 6 rows 22 inches apart.

3.1 Methodology

The proposed solution, assumes that a high altitude flight has covered the entire field and has revealed candidate areas where the plants potentially exhibit deficiencies. Methodologies for the detection of generic stressed areas based on pixel coloration are quite common in the area of RS with commercial solutions being readily available. The GPS locations of those areas are transmitted to a small-scale UAV in the form of way-points for a semi-automated low altitude flight. During the low flight, high resolution RGB images of the stressed areas are collected with hardware/software solutions that are commercially available and will be further discussed in the subsequent sections. The collected image sequences are provided as input to our methodology to undergo further processing.

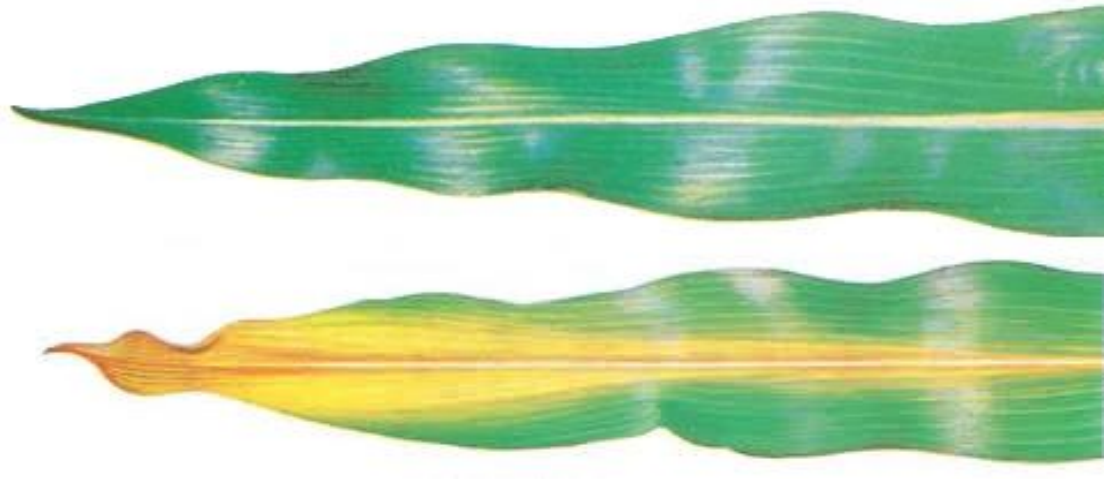


Figure 3.3: Healthy corn leaf (top) in comparison with a nitrogen deficient corn leaf (bottom).

Although the problems of leaf segmentation based on color and deficiency classification have been studied separately, to our knowledge there has not been a study that describes a pipeline looking at the plant pathology assessment problem holistically. In this work, we attack both the detection and classification problems and propose a complete solution that elevates simple RGB data into meaningful information for the farmer.

The proposed scheme consists of two parts, (i) the detection of Regions Of Interest (ROI) from the provided images, and (ii) the assessment of Ndeficiency inside the ROI. The pseudocode for the whole pipeline can be found in Algorithm 1. The input of this pipeline (lines 2-5) is an image of the corn plants, and three SVM models pre-trained to address the separation of green pixels, yellow pixels from soil, and the final assessment for the presence of Ndeficiency, as will be discussed in the following sections. The provided algorithm targets the accelerated execution run-time and without loss of generality, the assumption is made that one image will be partitioned into non overlapping segments (lines 6-7). Inside the main loop, lines 13-19 describe module (i) with;

- the enhancement of the colorspace for better separability through an image decorrelation technique [63],
- the extraction of superpixels [22] that guarantee the smoothness of the clusters and the speed-up of the clustering process,

- the creation of the combined colorspace queries $\mathbf{p}_j$,
- the Kmeans clustering that groups superpixels with similar chromaticity,
- the SVM classifier assigning the correct label to the green group of pixels g_g,
- the separation of the rest of the pixels g_r into the yellow pixel cluster g_y and soil, and finally
- the thresholding operation that selects yellow clusters g_{sy} above a provided pixel size.

Consequently, lines 20-26 display the implementation of module (ii). For each one of the g_{sy}, a bounding box is added surrounding it and this candidate box is subject to the classification that will make the final assessment of whether a smooth yellow cluster g_{sy} shows Ndeficiency or not. If the candidate is found to exhibit the particular deficiency, the counter of the deficient leaves will increase by one, otherwise the next candidate will be examined until all recommended regions are exhausted. The final result is the number and the location of Ndeficient leaves in the provided image.

3.2 Recommendation Scheme

The recommendation scheme presented in this section is responsible for the preprocessing of the input images and the extraction of ROI which visualize potential N-deficient candidates. The role of this scheme is vital for the overall process, since it greatly limits the candidate regions of interest when compared to a random creation of ROI over the analyzed images. In that way, the classification step that follows has a less computationally intense task to handle.

The concept behind a recommendation algorithm is the selection of meaningful subsets of pixels from a given image and this work identifies such subsets following the exhibited homogeneity of their color spaces. The conceptual flow of this recommendation module is found in Fig. 3.4 and can be separated in the following three steps:

1. a combination of unsupervised and supervised pixel grouping that produces two major clusters separating the green pixels from the rest,

2. a semi-supervised classification that extracts the yellow pixels, and

3. a low level morphological operation which eliminates noisy clusters assisting in the refinement of this module's output.

Pixels are clustered into three groups based on color information. The first group consists of green pixels which are associated with the green parts of the depicted plants, the second of yellow pixels, which correspond to potential N-deficient segments, and third, pixels that correspond to the soil. A diverse experimentation with pixels' intensity distributions in different color spaces indicated that unsupervised clustering methods are capable of separating the green parts of the plants from the rest of the image.

Initially, we aim to segment the green parts of the plants by employing an unsupervised two-step clustering scheme and a semi-supervised classification technique. These three steps include (i) a grouping technique that creates superpixels, (ii) a clustering step that breaks down the image into multiple groups of similarly colored pixels, and (iii) a 2-group classification that separates the green parts of the plants from the rest of the image.

The generation of SLIC superpixels is an unsupervised preprocessing step that significantly reduces the complexity of the subsequent clustering. At the core of SLIC is the K-means algorithm acting on the five dimensional vectors created by the concatenation of pixel coordinates and the L*a*b* colorspace;

$$\mathbf{s}_i = [\mathbf{x}_i \ \mathbf{y}_i \ \mathbf{L}_i \ \mathbf{a}_i \ \mathbf{b}_i]. \tag{3.1}$$

Due to the limited search space for each cluster center, as imposed by the SLIC algorithm, its complexity is linear to the number of pixels in the image $\mathcal{O}(n)$. This property combined with its innate characteristic to produce smooth pixel groupings makes it a powerful preprocessing tool.

Undoubtedly, SLIC with the K-means at its core holds the advantage of an unsupervised methodology that is data-driven and much more versatile than a purely supervised classification method which, apart from the cumbersome step of training the classifier, inherits the danger of over-fitting.

The next two steps are concluded in the feature space that results from the concatenation of the RGB and L*a*b* pixel intensities. This generates a feature vector $\mathbf{p}_i \in \mathbb{R}^6$

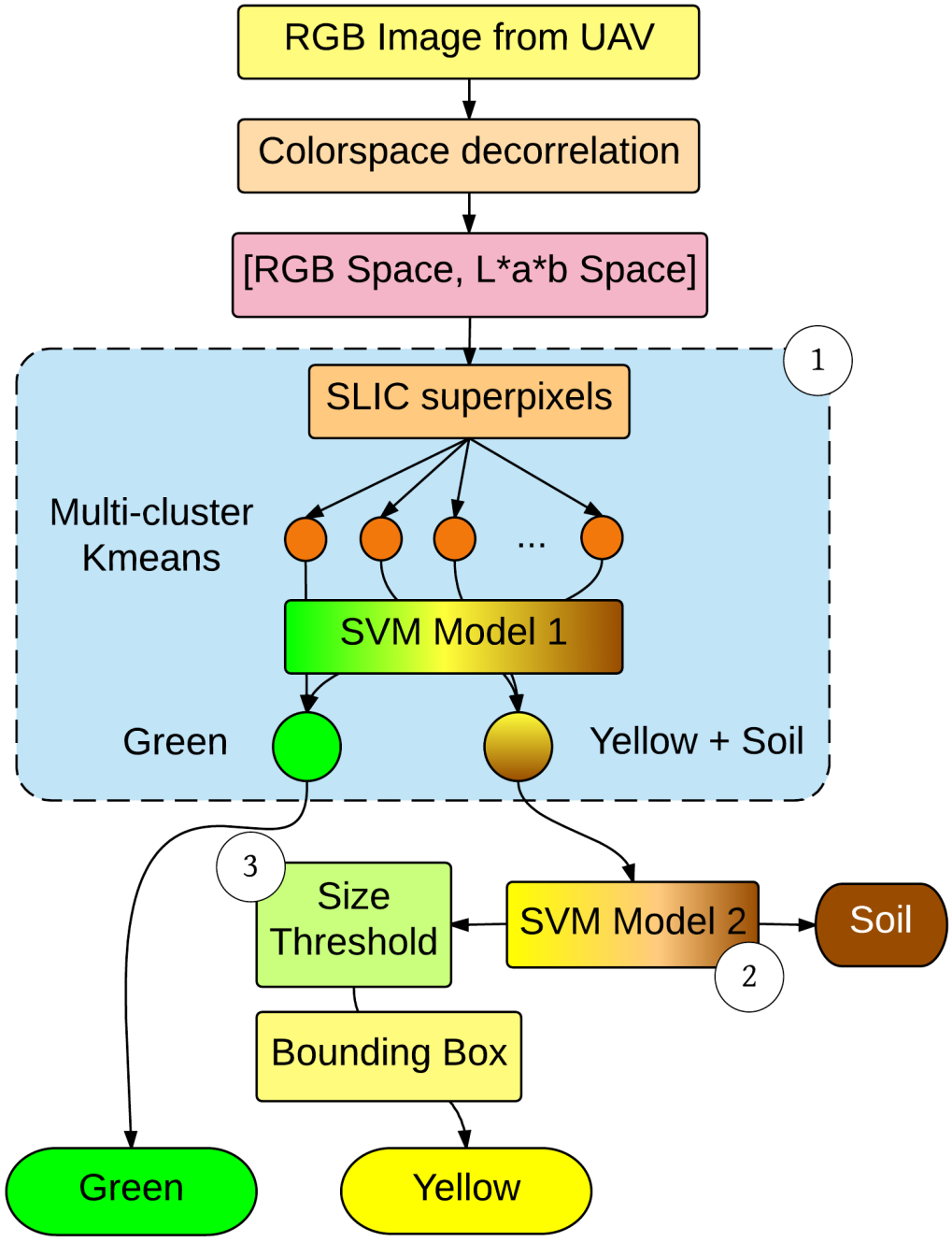


Figure 3.4: Flow chart of the Recommendation Scheme.

for pixel $i \in \{1, 2, ..., n\}$ as presented in Eq. (3.2). The selection of this representation was based on the theory proposed in [21] regarding the *essentially different* color spaces, as well as the recorded system's performance on a diverse subset of images captured during various illumination conditions. The **R**, **G**, and **B** represent the basic colors, **L** is the representative of intensity, and the **a**, **b** channels represent the normalized *rgb* colorspace.

$$\mathbf{p}_i = [\mathbf{R}_i \ \mathbf{G}_i \ \mathbf{B}_i \ \mathbf{L}_i \ \mathbf{a}_i \ \mathbf{b}_i] \tag{3.2}$$

For the selection of the appropriate clustering algorithm for the second step, we considered attributes such as the complexity with respect to the number of queries and the utilization of physical memory during the computation process. The K-means algorithm, a popular unsupervised clustering framework, met the requirements for the task while providing satisfactory performance. The computational complexity of K-means is $\mathcal{O}(Ikmn)$, with I the number of iterations until convergence is achieved, k the number of clusters, m the dimensions of the data space, and n the number of queries [64]. For the set of superpixel means $\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_n$, K-means computes clusters $g_1, g_2, ..., g_k$ by minimizing the objective function described in Eq. (3.3). Centroids of clusters μ_c are the mean vector of pixels belonging to cluster g_c.

$$\mathcal{J}_{Kmeans} = \sum_{c=1}^{k} \sum_{\mathbf{p} \in g_c} \|\mathbf{p}_i - \mu_c\|_2^2 \tag{3.3}$$

The K-means clustering is acting on the mean color values of the superpixels generated by the SLIC preprocessing step. Therefore, the clustering queries are the mean color values of each superpixel in the form of Eq. 3.2. This means that the m and n variables take the values 6 and ns respectively and since I, m, and n are known apriori, K-means for this application is linear only on the number of clusters k.

This unsupervised clustering scheme is performing well on grouping pixels with similar color values while being solely data-driven, but is unable to consistently assign the green label to the green pixel groups. This limitation comes from the random initialization of the cluster means. To overcome this limitation, a pre-trained supervised Support Vector Machine (SVM) model with a linear kernel was employed as the third

step to separate the clusters into two groups, one capturing the different variations of observed green colors (healthy leaves and stem), and a second describing the color variations of brown colors (soil, tassels, dry and deficient leaves).

SVM models are supervised learning models that are the result of minimizing for the constrained optimization problem described in Eq. (3.4). For the training of the SVM model the popular libSVM library[1] was utilized.

$$\begin{aligned} &\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i \;\; \text{subject to:} \\ &y_i(\mathbf{w}^\top \mathbf{p}_i + b) \geq 1 - \xi_i \\ &\xi_i \geq 0, \quad \forall i \end{aligned} \tag{3.4}$$

where, $\mathbf{w}$ and b are the parameters describing the learned hyperplane, ξ_i are introduced slack variables that account for the non-perfect separation between classes which is present in most realistic problems, and $y_i \in \{-1, +1\}$ describes the class labels of the training samples with (-1) corresponding to pixels depicting the soil and $(+1)$ depicting yellow pixels.

Once the green components of the plants are separated from the rest of the image, a second pre-trained supervised classification scheme partitions the non-green pixels into yellow (potentially deficient) and soil clusters. In the visible spectrum, the automated distinction between yellow pixels and pixels belonging to the soil was shown to be particularly challenging and a more refined strategy was devised for their separation.

The developed process requires the user to draw a single rectangle around pixels that represent soil and another rectangle around yellow pixels which represent leaves with Ndeficiency. Consecutively, a SVM model with a linear kernel is trained on these selections and is used for the classification between yellow and soil in the rest of the images. It is worth mentioning that images from different fields and under different illumination conditions were providing satisfactory segmentation results using an SVM model trained on a single image.

[1]https://www.csie.ntu.edu.tw/ cjlin/libsvm/

In a last step, a supplementary morphological operation removes small groups of connected regions of yellow pixels based on a threshold that considers their size, as seen in the middle and right columns in Fig. 3.8. The threshold was manually selected through a trial and error process and can fluctuate depending on the resolution of the initial image and the desired granularity of the results. The surviving objects that emerge from these morphological operations guarantee high performance of the feature extraction step described in the next section.

3.3 Detection Elimination

The main focus of this step is to refine the set of ROI provided by the recommendation scheme and identify candidates that exhibit the targeted condition with high confidence. This additional inference distinguishes *N*deficient regions from regions depicting tassels, healthy leaves and stressed leaves that cannot be assessed due to the advanced state of their condition (e.g., dry leaves on the soil). For this task we developed a supervised learning scheme that completes the inference cycle of this work by providing a binary label to the candidate regions. Positive samples correspond to image regions containing *N*deficient leaves while negative samples depict the remaining possibilities.

In an effort to enforce uniformity over the set of rectangular images extracted by the recommendation scheme, we resize them to a fixed size of 100×100 pixels. The image size was selected based on a parameter tuning process by observing the overall performance on a small subset of the constructed dataset. Following that, we evaluated a wide variety of popular computer vision descriptors towards deciding about the most pertinent image representation for the classification task in hand. In particular, six different descriptors were evaluated; Vectorized Pixel Intensities (VPI), Normalized Color Histograms (NCH), Histograms of Oriented Gradients (HOG), Bags of Visual Words (BVW), Sparse Codes (SC), and Fischer Vectors (FV).

3.3.1 Vectorized Pixel Intensities

Vectorized Pixel Intensities is the simplest descriptor explored in this work. In particular, each color channel of a resized RGB image is vectorized to create three 10,000-dimensional vectors (one for each channel), which are then concatenated to form a

30,000-dimensional representation for the image. The high dimensionality of the descriptor along with its innate ability to capture color information render it as a strong candidate for the classification task in hand, yielding performance comparable to Normalized Color Histograms (NCH) assignment with an accuracy of 87.5% as also reported in Table 3.2. Surprisingly, expanding the color space by concatenating the L*a*b* information did not produce better results.

3.3.2 Normalized Color Histograms

Normalized Color Histograms were computed in an effort to exploit the distinct yellow color associated with *N*deficiency. Different parameter configurations were tested in a trial and error process in order to conclude with a high performing configuration. In particular, we computed six histograms of 50 bins, each corresponding to one color channel (R, G, B, L*, a*, b*). After histograms are normalized to sum up to one, they are concatenated producing an image descriptor of 300 dimensions. As observed in Table 3.2, NCH performs significantly better than most of the explored descriptors ranking as the top-performing descriptor. The classification results achieved 90.6% accuracy.

3.3.3 Histograms of Oriented Gradients

Histograms of Oriented Gradients [65] were introduced in the Computer Vision community for the detection of humans. The deployment of the HOG descriptors in this study was driven by the observation that *N*deficient leaves exhibit a very distinct V-shaped pattern as illustrated in Figure 3.3. In an effort to capture this distinguishing feature, HOG descriptors were computed. Images where divided in non-overlapping patches of size 10×10 pixels and intensity gradients were computed on each patch. Gradient histograms of 50 bins were then computed for each patch and concatenated to produce a 15,400-dimensional descriptor for the whole image. HOG, based on our results, appears as averagely performing descriptor ranking in the fourth place with 72.5% accuracy. For our experiments we utilized the VL-FEAT toolbox [2] to compute the HOG descriptors.

[2] http://www.vlfeat.org/

3.3.4 Bag of Visual Words

Capitalizing on local point features and their descriptors, Bag of Visual Words summarize their information content via the computation of a constructed code-book [66]. In this study we create a code-book on Scale Invariant Feature Transform (SIFT) descriptors. In particular, K-means clustering algorithm is used to cluster the computed descriptors to k clusters. The centroid of each cluster behaves as a codeword in the computed codebook. For our experiments we use 2,000 images for constructing a codebook of 100 codewords. For every image, after its SIFT descriptors are extracted, an encoding is computed based on the frequency these descriptors are assigned to every cluster (codeword). This information is summarized in one histogram of 100 bins which is then used for the classification process. The performance of BVW was the lowest reaching an accuracy of 65.8%.

3.3.5 Sparse Coding

Sparse Coding (SC) utilizes Dictionary Learning (DL) techniques [67]. Each query image is split into smaller patches which are used to create three overdetermined matrices whose columns represent basis vectors of a high dimensional space. These matrices, each for one of the R,G, and B colors, are used to extract the reconstruction residual for each of the small patches of each image in vector form. All the means of those residual vectors form the Sparse Coding; the query for the classification model. With a dimensionality of 180, SC achieved 76.9% accuracy. For the experiments we employed the SPAMS toolbox [3] to compute the SC descriptors.

3.3.6 Fischer Vectors

The last image descriptor considered in this study was the Fischer Vector [68] which, similarly to the BVW, is an encoding of local point feature descriptors. The main difference between FV and BVW is that K-means is substituted by Gaussian Mixture Models (GMM). For our experimental evaluation we computed a clustering assignment using 50 Gaussians which yielded a 12,800-dimensional descriptor for an accuracy of 72.2%.

[3] http://spams-devel.gforge.inria.fr/

Using the aforementioned descriptors, an effort to identify the Ndeficient rectangles of the recommendation scheme was performed. We accomplished this by training an SVM model in a manner similar to the formulation described in Eq. (3.4). It should be noted that rather than working with feature vectors at a pixel level ($\mathbf{p}_i$), we train our model at an ROI aggregate level. Letting $\mathbf{d}_i$ denote any of the aforementioned descriptors of this section, the first set of inequality constraints becomes $y_i(\mathbf{w}^\top \mathbf{d}_i + b) \geq 1 - \xi_i$. In addition, positive samples now correspond to N-deficient samples while negative samples depict samples not containing deficient leaves.

3.4 Experimental Setup

Figure 3.5: Our two robotic aerial platforms. *Left:* MikroKopter OKTO XL, *Right:* DJI Matrice 100.

The collection of high resolution imagery in the visible spectrum took place over the course of four cultivation periods (2014-2017) with data capturing a combination of growth stages varying from V5 to V12. Two maize fields in the state of Minnesota were selected for testing our methodology as also illustrated in Fig. 3.6. Two very different soil types were considered, namely, sandy and clay loam. The soil type greatly affected the number of visible deficient leaves during the early stages (V5 and V8). The clay loam soil tends to naturally preserve nutrients which are absorbed by its environment resulting in a normal plant development even without the application of fertilizer. The sandy soil, on the other hand, depends solely on the applied fertilizer exhibiting visible Ndeficiencies early on.

To demonstrate the independence of our approach from a particular hardware, the data were collected by two different platforms which are depicted in Fig.3.5; a

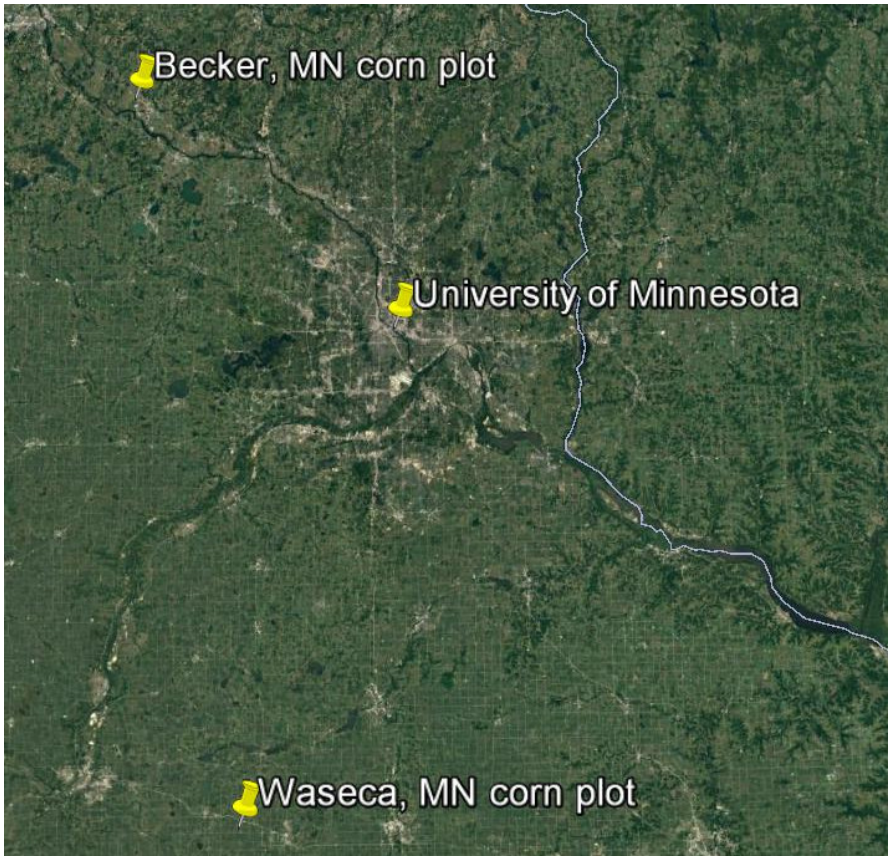


Figure 3.6: Satellite image depicting the locations of the two test fields in Becker and Waseca, MN. The images were generated using Google earth.

MikroKopter Okto XL equipped with a NIKON D7200 RGB sensor and a DJI Matrice 100 with a Zenmuse X3 and a Zenmuze Z3 RGB high resolution camera.

Over 800 static images such as the ones seen in Fig.3.10 were collected at the growth stages V5, V8, and V12 from plots that were treated with 6 different Nfertilizer densities (0, 60, 120, 180, 240, and 300 lb/ac). Out of these, the images that were showing solely healthy grown plants were discarded; in a real world scenario we are interested in identifying Ndeficiencies only on stressed areas of the field. The resolution of the images varies within a range of 0.2 to 0.47 cm/pixel.

3.5 Algorithmic Results

Segmenting the green parts in a single color space is not accurate for all images. This is especially true in cases with few Ndeficiencies, where a significant variance in the representation of the green color is present such as the sample image on the bottom left of Fig. 3.7. The accuracy of the segmentation increases when combining the clustering results of the two color spaces. This method achieves robust results in the segmentation of green pixels for all the subject images.

In the first module, following the proposed methodology the green cluster is created with an average accuracy of 96.3%. A high granularity in the superpixels step increases the detail of the clustering, but might create several disconnected artifacts which could introduce over-segmentation later in the process. Depending on the resolution of the images, a typical number of superpixels ns is around 5,000. Demanding a low number of clusters k in the K-means clustering step has shown acceleration while maintaining high accuracy for different illumination cases, with a typical number of 15 clusters being sufficient.

Figure 3.7: Green segmentation results for two sampled cases of different illumination and significant green color variance. *Left:* original images, *Center:* groundtruth, *Right:* segmentation result in the decorrelated and stretched RGB color space.

Next, the distinction between yellow areas and soil is performed by an SVM classifier. The trained SVM model is sensitive enough to pick individual yellow pixels, as seen in the middle column of Fig.3.8 and the thresholds on the morphological operations can be customized by the user to match the desired granularity of the results. The resulting ROI for a sample of images in different scales and illuminations can be seen in Fig.3.10 and the execution time of the module divided into its core components is found in

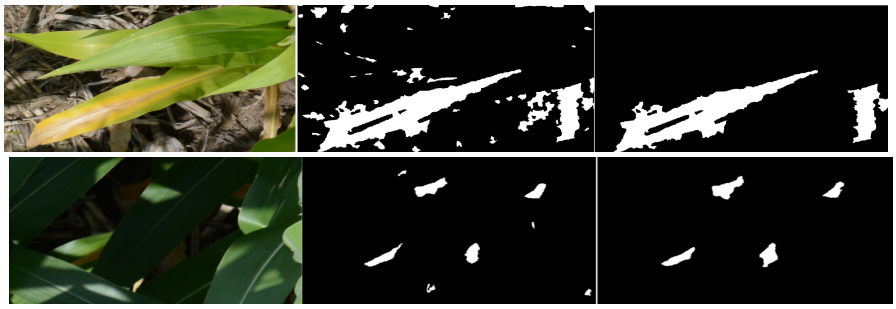


Figure 3.8: Yellow segmentation results for two sampled cases of different illumination and significant color variance. The results were produced using the same SVM model which was trained in one completely different image. *Left:* original images, *Center:* yellow pixels as clustered by the SVM model, *Right:* yellow areas of interest after the morphological operations.

Table 3.1.

For the second module that involves the elimination of ROI that do not exhibit *N*deficiency, the results of the 6 descriptors show the superiority of NCH over the rest with performances that achieve accuracy, sensitivity and specificity reaching 90.6%, 90.6%, and 90.7% respectively.

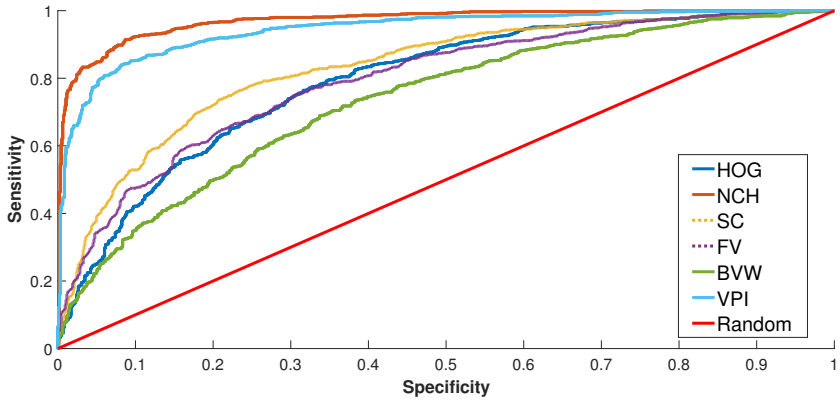


Figure 3.9: ROC curves for the classification results of the 6 descriptors. The random line is provided for reference.

The results, as portrayed in Table 3.2 and Figure 3.9, validate the hypothesis that the distinct yellow color of the deficiency is captured by the NCH descriptor. It is rather interesting to notice the differences in performance between features that capture color versus the ones that capture texture information. From the results it is apparent that color information captured by NCH is more distinctive than texture information as

expressed by HOG and BVW, or even their mix as captured by Sparse Coding. The final result of the whole pipeline with the NCH classifier is depicted in Fig.3.11.

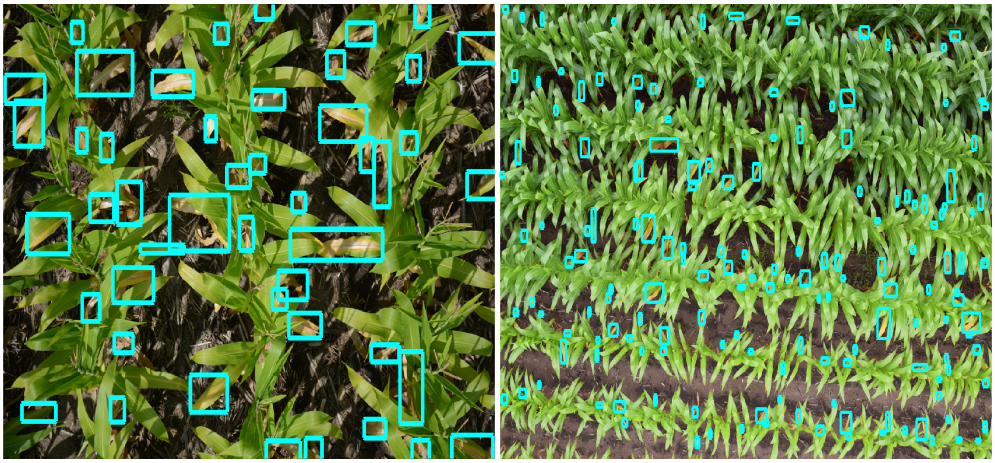


Figure 3.10: Results of the recommendation scheme on corn images with different resolution and Nconcentration. The cyan boxes represent the ROI that will be used as an input to the Ndeficiency assessment module. It is interesting for further investigation that the methodology considers the tassels of the corn plants as areas of interest due to their color.

3.6 Transferability

One important aspect of our system is its transferability to other crops and deficiencies that show similar patterns as the Ndeficiency in corn. As long as the deficiency is expressed with yellowness, the same algorithms can be used for training and prediction of the stress. In Figures 3.12a and 3.12b, our recommendation algorithm has been applied to randomly selected images found on the internet representing $N$and Iron Chlorosis deficiencies on soybean.

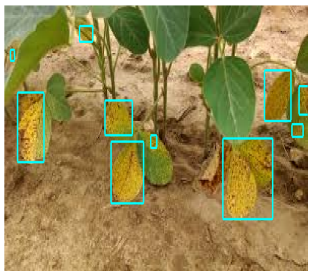
3.7 Limitations

When provided with appropriate images, the current methodology is able to handle the majority of visible Ndeficiency cases in corn plants, but still there are some limitations. As seen in Fig. 3.13a, the high density of the canopy occludes the deficiencies that usually are most prominent at the lower leaves. Further, the pipeline is not completely

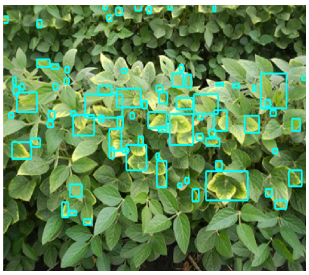
Figure 3.11: Results of the detection elimination scheme on a series of stitched images depicting two adjacent fields with different Nconcentrations. On the left part of the image, the corns were not treated with fertilizer (0 lb/ac) while on the right the corn plants received the maximum fertilizer amount (300 lb/ac). All the bounding boxes resulted from the recommendation scheme. The red boxes represent the true positive detections and the cyan boxes represent the false positive detections according to the SVM model trained on the NCH features. The whole stitched image was broken down to 16 parts for the processing and was reassembled after the termination of the algorithm.

invariant to the dataset provided; it requires training of the SVM models and tweaking of the k, ns and th parameters depending of the resolution of the images, the targeted size of the deficiency and the plant types.

For the final classification process in the second module, the limitations are introduced by the ambiguous nature of the problem which introduces human errors in the annotation. In an image some leaves exhibit Ndeficiency and the pattern, as introduced in Fig. 3.3, is clearly visible. On the other hand, there are leaves that are partially occluded with only a small yellow part visible, or others that have lost their green color due to a severe stress. The characterization of a leaf as deficient when it is not entirely visible, is subjective and depends on the annotator as seen in Figure 3.13b. Several times, even the context affects the annotator's decision since a yellow leaf part has a high chance to belong to a Nstressed leaf due to its neighbouring leaves exhibiting that particular stress. To address this limitation, the dataset for the classification has been labeled by four expert agronomists under the assumption that at least a part of the characteristic V-shape is visible.



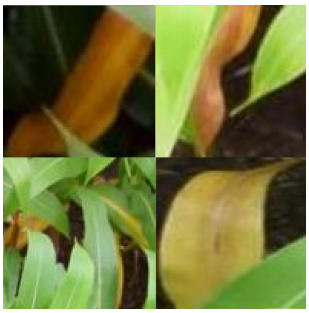
(a) *N*deficiency on soybean plants.



(b) Iron Chlorosis deficiency on soybean plants.

Figure 3.12: Transferability of the recommendation algorithm to different plant types, stresses, and viewpoints.

(a) The *N*deficiency that was not detected is enclosed within a yellow box.



(b) Oblique cases of *N*deficiency. The annotators did not have a clear answer in cases when the "V" shape is not in the picture, the deficiency was too severe, or was not the actual focus of the image.

Figure 3.13: Two images that visualize limitations in our system.

3.8 Practical Application

The proposed methodology is agnostic to the robotic platform that acquires the panchromatic images but, based on the current market solutions, we assume that the platform is a UAV associated with a software package that allows setting up the flight trajectory to cover the whole field under some height and speed constraints. The detection of N deficiency requires a high amount of imagery detail, therefore, a high altitude first flight is needed to detect the areas that appear stressed. A second flight in lower altitude will capture detailed images that will reveal the cause and severity of the stress.

Most professional farmers maintain several hundreds of acres and usually are not willing to perform two lengthy flights for the detection of N, unless they are convinced for the financial returns. The role of this inspection is usually undertaken by consultants that perform the surveillance and propose a treatment for the specific field. These consultants are the targeted group of this application.

Assuming a UAV with a typical operational time of 25 minutes, the first flight can be performed in a few minutes at a 400 feet altitude and can provide a good visual understanding of the field status. The areas that show stress will be visited sequentially at a 50 feet altitude and only a small number of images will be collected. The low altitude flight depends on the characteristics of the mounted camera but for a high resolution camera a maximum of 8 rows of corn should be visible. The collected images will be processed by the algorithm and for each image the N deficient leaves will be pointed out. This information can then be consumed by the consultant in a model for the estimation of the N fertilizer that is needed.

Algorithm 1: Pseudocode of the proposed methodology.

Result: N = number and locations of N-deficient ROI in image

Initialization:
$\mathbf{im}$ = input image
$[\mathbf{w}_1, b_1]$ = SVM model to assign label to green clusters
$[\mathbf{w}_2, b_2]$ = SVM model to separate yellow from soil
$[\mathbf{w}_3, b_3]$ = SVM model to assess Ndeficiency
z = number of segments to partition image
$\mathbf{I} = \mathbf{partition}_z(\mathbf{im})$
ns = number of superpixels
k = number of clusters in Kmeans
th = threshold on the size of ROIs

Main Loop:
for $i = 1 : z$ **do**
  $\mathbf{I_d} = \mathbf{ImageDecorrelation}(I(i))$
  $\mathbf{I_s} = \mathbf{SLIC}(\mathbf{I_d})$
  $\mathbf{p}_j = [\mathbf{R}_j\ \mathbf{G}_j\ \mathbf{B}_j\ \mathbf{L}_j\ \mathbf{a}_j\ \mathbf{b}_j], \quad \forall\ j\ \in\ \mathbf{I_s}$
  $c_{1,\ldots,k} = \mathbf{Kmeans}(\{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_{ns}\},\ k)$
  $[g_g, g_r] = \{\ l\ \in\ g_g :\ sign(\mathbf{w}_1^\top \mathbf{c}_l + b_1) > 0\ \}$
  $g_y = \{\ l\ \in\ g_r :\ sign(\mathbf{w}_2^\top \mathbf{p}_l + b_2) > 0\ \}$
  $g_{sy} = \mathbf{Threshold}(g_y, th)$
  for $j = 1 : |g_{sy}|$ **do**
    candidate = $\mathbf{AddBoundBox}(g_{sy}(j))$
    $\mathbf{d} = \mathbf{ComputeDesc}$(candidate)
    if $sign(\mathbf{w}_3^\top \mathbf{d} + b_3) > 0$ **then**
      N++
    end
  end
end

Component (i/ii)	Average execution time (ms)
(i) Green clustering	233.62
(i) Yellow clustering	28.81
(i) Morphological operations	2.75
(ii) Class prediction on entire image (NCH)	11.10
Total time (800x1200)	276.28

Table 3.1: Average execution time for the core components of the recommendation scheme (i) and the *N*deficiency assessment (ii) on an image of size 800x1200 pixels. The processing was done in MatLab 2016a on a 2.3GHz i7-2820QM processor.

Descriptor	Accuracy	Sensitivity	Specificity	AUC
VPI	87.5%	87.7%	87.2%	0.95
NCH	**90.6%**	**90.6%**	**90.7%**	**0.97**
HOG	72.5%	72.0%	73.1%	0.79
BVW	65.8%	67.9%	63.8%	0.71
SC	76.9%	77.2%	76.7%	0.83
FV	72.2%	76.2%	68.0%	0.80

Table 3.2: Detailed classification results for the 6 different image descriptors. The accuracy, sensitivity, specificity, and area under the curve are provided. The dataset consisted of 9,021 suggestions split into 4,363 positive (*N*deficient) and 4,658 negative samples.

Chapter 4

3D Segmentation of Corn Plants

A review of the existing literature in PA, as presented in Sec. 2.3, indicates the need for accurate and frequent plant biometrics estimation. An accurate model of the canopy can reveal important information regarding the state of crops and provide feedback to growth models. For agricultural applications, measurements such as the biomass or the angle of the leaves with respect to the stem are powerful indicators connected to the crops' health, growth state, and ability to photosynthesize efficiently.

In contrast to existing invasive methods for accurate biomass calculation that rely on plant deconstruction, their non-invasive alternatives are preferred in commercial applications since they leave the crops intact. These state of the art solutions approximate the real phenotype based on mathematical models and sparse measurements collected randomly throughout the field. Such approaches are inherently problematic since the average of a few measurements provides a single result that characterizes a wide area. The spatial sparsity in combination with the use of cumbersome sensors which are handled by a human impose significant constrains in the sampling process; collecting measurements for a large number of points over the field can be prohibitive, costly and inaccurate.

An automated methodology for providing detailed and reliable information from 3D models of corn canopies would directly address the needs of both researchers and companies interested in corn development. The next two chapters propose a methodology capable of estimating the phenotype of a group of plants using their 3D models. In the following sections we will explore a pipeline that is able to consume 3D reconstructions

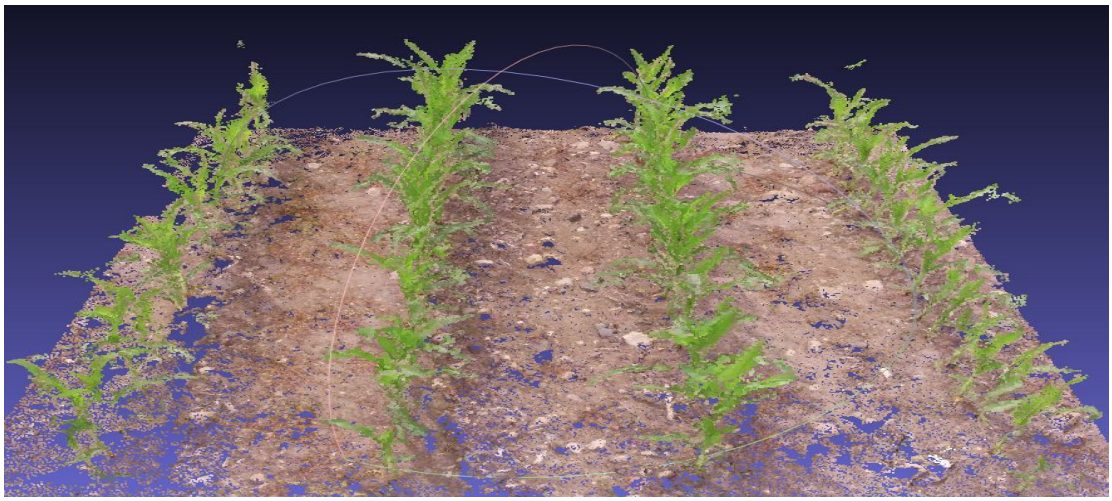


Figure 4.1: Example of a 3D model reconstructed by imagery. A four row segment of corn plants at "V5" growth stage.

of crops in the field and provide measurements with a granularity and frequency that have not been available to the agriculture community before.

When a 3D model such as in Fig. 4.1 is accessible, the surface of all of its leaves should be observable, overcoming the leaf occlusions and resulting in a more accurate volumetric information about the plant. The proposed methodology is focusing on maize of growth stages between "V3" and "V6" [4], when the plants are still susceptible to treatment and introduces the first attempt for a low-cost, mobile, and easily deployable solution for automated computation of the plant's phenotype. For the reader's education, Fig. 2.1 and Sec. 2.1 summarize the different growth stages of corn plants.

4.1 Methodology

This section provides an introduction to the basic framework used in this analysis and enumerates the main steps of the pipeline for the manipulation of the 3D reconstructions and the extraction of the phenotypic information. Following Fig. 4.2, the pipeline is subdivided in three main steps:

1. the 3D reconstruction that uses a Structure from Motion (SfM) software to transform the high resolution images to a 3D model,

2. the segmentation step that assigns labels to the parts of the 3D reconstruction and more specifically (i) the ground, (ii) the individual plants, and (iii) the stems and leaves of each plant, and

3. the computation of the phenotype that takes advantage of the segmentation step to combine the geometries of stems and leaves and extract measurements and statistics.

The core methodology for the segmentation as presented in this chapter requires a dense 3D reconstruction of a group of plants. This can be obtained by capturing high resolution images of the targeted group while moving in a circular fashion as seen in Fig. 4.3 and employing a 3D reconstruction toolbox such as [69]. Obtaining the 3D reconstruction is decoupled from its processing and does not impose any platform constraints – it only requires an RGB sensor and sufficient computation capabilities. It is possible to acquire the necessary imagery through handheld or UAV mounted cameras.

The extracted 3D reconstruction has the form of a collection of points, known as a *point cloud*, P. Each point p of the point cloud has a physical representation in 3D space and is expressed by a vector of three values along the x, y, and z directions. Associated with each p are three more values that reflect the r, g, and b chromas of the red, green, and blue channels. Consecutively, each point of the point cloud can be represented as:

$$p = [x, y, z, r, g, b]. \tag{4.1}$$

The complexity of the outdoor crop field is apparent in the obtained 3D object 4.1 and requires an initial segmentation step able to separate the complex scene into simpler components. We define three main semantic categories that are used as building blocks for any phenotype extraction; the soil, the individual plants, and their respective stems and leaves.

The soil represents the background and usually takes a significant portion of the generated 3D model. It is useful for the separation between plants and facilitates the alignment of the point cloud along the z-axis. It is vital to separate each crop individually because the goal is to provide information on a plant level, and because accessing individual crops benefits the subsequent steps of separating stems from leaves. The

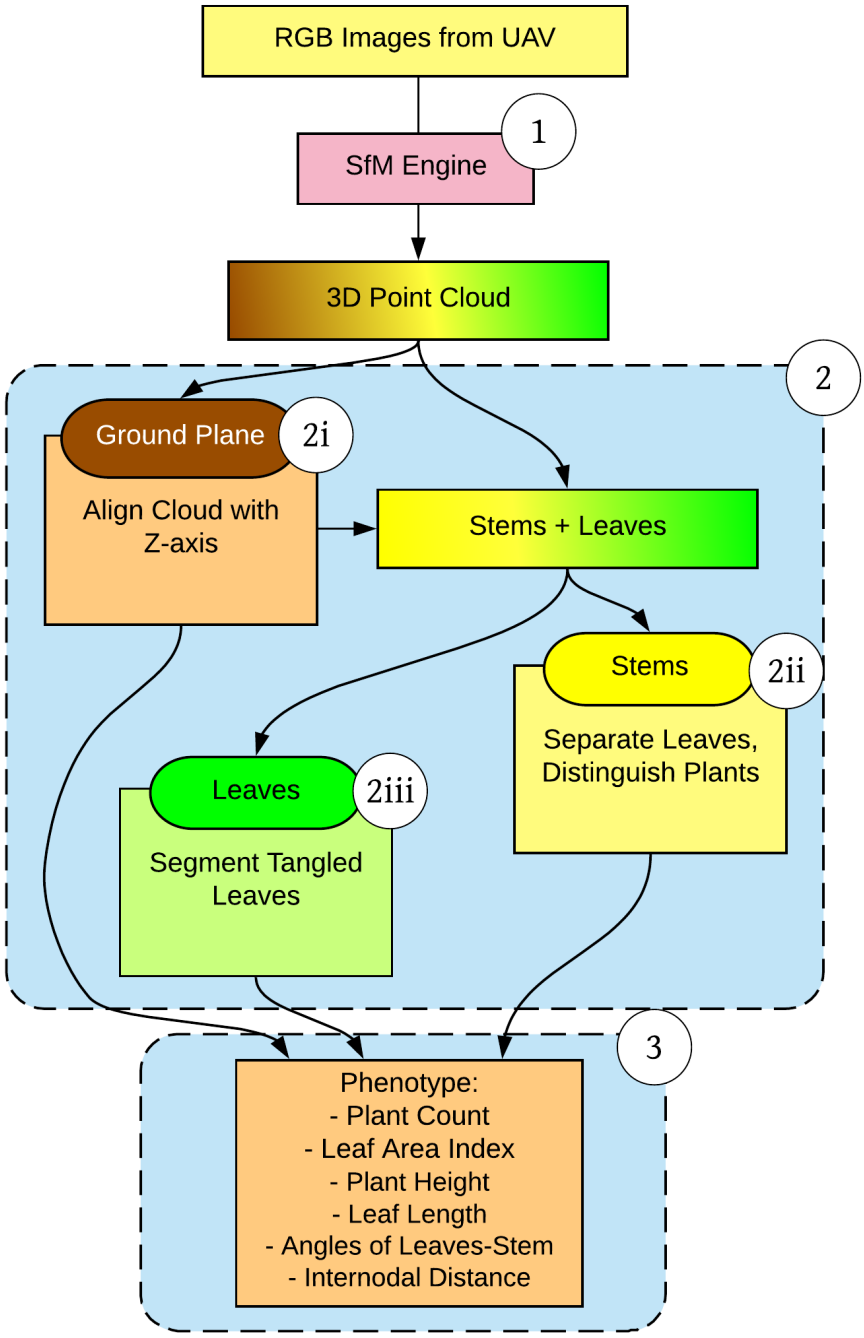


Figure 4.2: The diagram visualizes the proposed pipeline for the segmentation and biometrics extraction from a set of images acquired by a camera mounted on a UAV.

stem is the skeleton of each plant, holds information on the number of plants, their row spacing and height, and it is the connecting element between all the leaves of the same plant. Finally, the leaves are the main semantic elements of our analysis since they comprise the majority of the biomass of a plant and reveal its ability to photosynthesize. The following sections present the algorithms that constitute the segmentation pipeline.

4.2 Preprocessing

Each time a 3D reconstruction is provided, a series of steps transform it into an input that is compatible to the manipulation pipeline. These transformations attempt to alleviate the inherent shortcomings of the SfM.

4.2.1 Scaling

The 3D reconstruction resulting from an SfM algorithm is up-to-scale equivalent to the actual scene, which means that any geometric computations performed on the reconstructed scene are not directly comparable to real world measurements. Furthermore, this discrepancy is different for every 3D reconstruction forcing the selection of different algorithmic constants at each execution. Our solution in determining the scaling factor s of the reconstruction in order to correct the geometric inconsistencies and verify the correctness of our computations is to compare the inter–row distance of the real world corn plants h_{real} against the inter–row distance of the reconstructed corn plants $h_{reconstructed}$. This scaling ratio is applied to the whole point cloud and is computed as:

$$s = \frac{h_{real}}{h_{reconstructed}}. \tag{4.2}$$

The inter–row distance was chosen as one of the most reliable constants in a field; during the planting, farmers select this distance and the tractors are configured to accurately follow it. As seen in Fig. 4.4, the user is prompted to measure the distance between rows with a 3D interactive tool and this $h_{reconstructed}$ value is used to initialize the scaling. This interaction needs to take place for each new reconstruction.

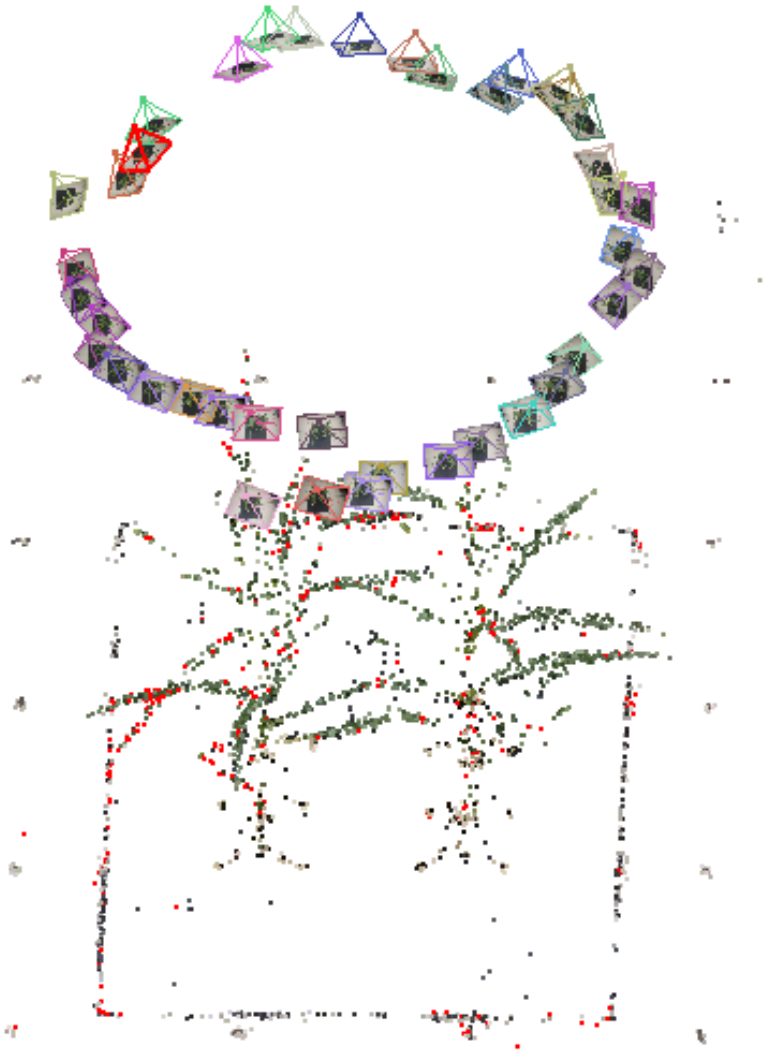


Figure 4.3: The sparse reconstruction resulting from the VisualSFM software. In this non-limiting example, several high resolution images were using a handheld camera while moving in a circular fashion around six artificial corn stalks.

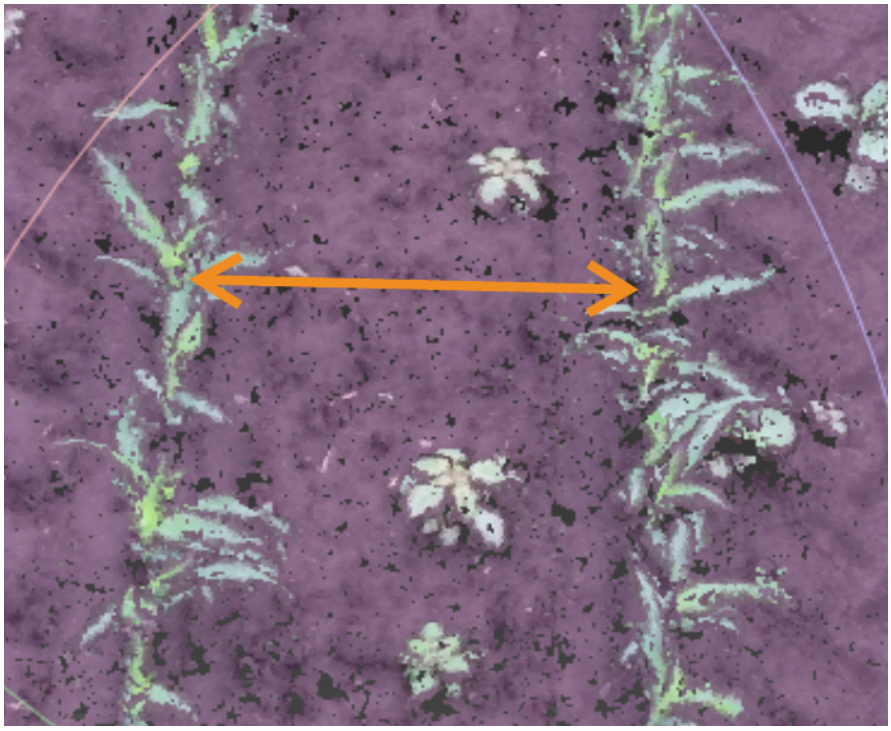


Figure 4.4: The user is prompted to use an interactive tool such as MeshLab [1] to measure the inter–row distance of the reconstruction. The real world distance is usually 22 or 30 inches for Minnesota.

4.2.2 Point Cloud Alignment

When the plant stems assume a position nearly perpendicular to the x-y plane and the ground is translated to match the x-y plane, several geometric priors can be utilized efficiently to reduce the complexity of the subsequent steps. That is why, after the scaling we wish to rotate the point cloud perpendicular to the z-axis and locate its ground plane.

Initially, we make use of the color information of each point and perform a color clustering that separates the brown from green points using the algorithm described in 3.2. This step results into two point clouds, one holding mostly ground points P_{gr} and the other mostly vegetation P_{veg}. This clustering step allows the undisturbed processing of ground and vegetation clouds separately and eliminates problems created by artifacts such as rocks or significant difference in elevation. The P_{gr} is first processed to acquire a good estimate of the ground plane.

Thankfully, the points belonging to the ground plane are the majority of P_{gr} and are easily detectable by a RANdom SAmple Consensus (RANSAC) [70] designed to estimate the coefficients of a 3D plane. The core of the RANSAC algorithm for the estimation of the ground plane P_{pl} utilizes the simple linear model:

$$\begin{aligned} ax + by + cz + d = 0 \\ \mathbf{n}^T\mathbf{x} = -d, \end{aligned} \tag{4.3}$$

with $\mathbf{n} = [a\ b\ c]^T$ and $\mathbf{x} = [x\ y\ z]^T$, and solves for the normal $\mathbf{n}$ through the covariance matrix $C \in R^{3x3}$ as computed by the set of plane inlier points $S \in R^3$:

$$C = \sum_{i=1:|S|} (s_i - \hat{s})(s_i - \hat{s})^T, \tag{4.4}$$

where $\hat{s} \in R^3$ is the mean of all $s_i \in S$.

The covariance matrix C captures the dispersion of the ground points and its three singular vectors that can be computed by its singular value decomposition (SVD), describe the three main directions of this dispersion. Since the plane is a flat surface, the normal $\mathbf{n}$, which is perpendicular to the plane, indicates the direction with the least variance and is captured by the singular vector corresponding to the smallest singular

value. After the acquisition of $\mathbf{n}$, d is directly computed from Eq. 4.3 by substituting $\mathbf{x}$ with $\hat{s}$ which is a good representative for the points belonging to the plane.

At this stage, we wish to align the normal $\mathbf{n}$ of the ground plane with the normal of the x-y plane $\mathbf{k} = [0\ 0\ 1]^T$. The solution is utilizing the Rodrigues' rotation formula, where the point cloud is rotated around an axis $\mathbf{u} = cross(\mathbf{n}, \mathbf{k})$ perpendicular to both $\mathbf{n}$ and $\mathbf{k}$ by the angle between them $\theta = angle(\mathbf{n}, \mathbf{k})$:

$$R = I + \sin(\theta)\lceil \mathbf{u} \rfloor_{\times} + (1 - \cos(\theta))\lceil \mathbf{u} \rfloor_{\times}^{2}, \tag{4.5}$$

where $I \in R^{3x3}$ is the identity matrix and $\lceil x \rfloor_{\times}$ is the skew symmetric matrix form of the vector x. The rotation matrix $R \in SO(3)$ is acting on all the points in the point cloud P and the resulting point cloud is aligned perpendicularly to the z-axis. Finally, a mean of the P_{pl} is computed and subtracted from all the points in P to move the ground plane on top of the x-y plane.

Attention is required by the sign of $\mathbf{n}$ so that the rotation is successfully orienting the point cloud. A sanity test is performed as a last step by checking all the z values of the points and if the majority of them is found positive the solution is accepted, otherwise the process is repeated with $\mathbf{n} = -\mathbf{n}$.

4.2.3 Noise Filtering

As in most real world data, the noise in the data is present with the form of uncertainty in the 3D measurements and 3D artifacts forming undesired artifacts (e.g. stones in the field). One step after the separation of the main point cloud P into P_{gr} and P_{veg}, the second is treated with a custom filtering algorithm similar to DBSCAN [71] that removes clusters of size less than a threshold n if they are located further away than a distance d from any other point cluster.

The details of this $\mathcal{O}(n \log n)$ algorithm are seen in Algorithm 2 where a KDtree structure is selected for fast ($\mathcal{O}(\log n)$) neighboring points search, a boolean vector *visited* keeps track of the points that have been visited, and another boolean vector *keepers* stores the indeces of the points that comply with the two aforementioned thresholds.

Algorithm 2: Filtering algorithm for the removal of noisy points and artifacts.

Result: ptsFiltered = set of points surviving the filtering
Initialization:
$kdtree$ = **createKDtree**(P_{veg})
$N = |P_{veg}|$
keepers = False($1 : N$)
visited = False($1 : N$)
Main Loop:
for $i = 1 : N$ **do**
    if ***visited***(i) **then**
        continue
    end
    $pt = P_{veg}$(i)
    neighborsIndex = **findNeighbors**($kdtree, pt, d$)
    if |**neighborsIndex**| $> n$ **then**
        keepers(**neighborsIndex**) = True
        visited(**neighborsIndex**) = True
    end
    visited(i) = True
end
ptsFiltered = P_{veg}(**keepers**)

4.2.4 Skeletonization

The extracted 3D reconstructions usually have a few million points and the processing power required to treat those is quite high. In an effort to reduce the running–time and add robustness against noise, the P_{veg} is undergoing a skeletonization step that thins out the excess points but retains the ones that express the basic topology of the plants Fig. 4.5.

Our custom skeletonization Algorithm 3 takes advantage of the alignment of the point cloud that was described in 4.2.2 and splits it into thin slices of height h along the z-axis. Each slice now contains k points of similar height z that are spread across the x-y plane and the goal is to perform a per–slice clustering to find points that best represent the rest. The clustering creates at least one cluster Cl at each slice and is performed utilizing a Euclidean Clustering technique ($\mathcal{O}(n \log n)$) described by Rusu [72] and the only parameter needed is a radius r defining a sphere for the 3D space search. The complexity of this algorithm is mainly the Euclidean Clustering that is repeated k

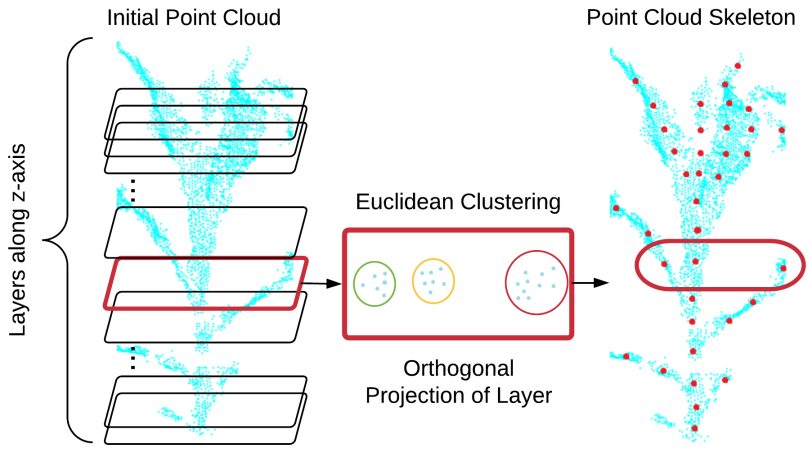


Figure 4.5: The thinning process of the 3D reconstruction of a single plant. The points are divided in layers based on their height and in each layer a euclidean clustering algorithm performs a grouping. The centroid representatives of each group form the final thinned cloud.

times, once for each slice, complemented by the computation of the cluster average to extract the representing point. This brings the total complexity to $\mathcal{O}(mk \log k)$ with m being the number of clusters found in a slice. An important element of this algorithm is the ability to associate points with their representatives and this is achieved by an index vector $associationIndex$ that stores the representative for each point.

4.2.5 Limitations

The preprocessing steps are meant to bring the initial point cloud to a usable form, but sometimes they are not enough. Starting with the scaling, the interactive selection of points on two consecutive rows is prone to human error. Next, the point cloud alignment is based on a plane detecting RANSAC algorithm which works well only when there are enough plane points. In growth stages of "V8" and above when the canopy is dense, very few ground points are reconstructed therefore the detection of the ground may be erroneous affecting the whole pipeline negatively. The noise filtering slows down significantly as more points are considered this is why dense canopies are avoided. Finally, the skeletonization may group together points that belong to different plants if they are close enough in a given z slice. These topics are open problems that

Algorithm 3: Skeletonization algorithm for the thinning of the vegetation point cloud.

Result: **skeleton** = set of points forming the skeleton
Result: **associationIndex** = vector storing the representative of each point
Initialization:
$N = |P_{veg}|$
$counter = 1$
$low =$ **minimumZ**(P_{veg})
$high =$ **maximumZ**(P_{veg})
$slices = low : high$
associationIndex = **zeros**$(1 : N)$
skeleton = $\varnothing$
Main Loop:
for $i = 1 : |slices| - 1$ **do**
    ptsIdx = **ptsBetweenSlices**$(i,\ i + 1)$
    $Cl =$ **euclideanSegmentation**$(P_{veg}($**ptsIdx**$),\ r)$
    for $j = 1 : |Cl|$ **do**
        $centroid =$ **average**$(Cl(j))$
        skeleton$(counter) = centroid$
        associationIndex$($**ptsIdx**$(Cl(j))) = centroid$
        $counter++$
    end
end

require further exploration to transform the proposed solution and expand it to denser canopy structures.

4.3 Individual Plant Segmentation

Point clouds that depict crop field scenes benefit greatly from the previously presented transformation and preprocessing schemes. The updated P_{veg} point cloud is now ready to be processed and the first step towards the final segmentation goal is the extraction of the individual plants. First, an original algorithm that takes advantage of the geometry of the corn plants is presented, followed by its application in the detection and separation of the individual plants.

4.3.1 Randomly Intercepted Nodes

The corn plants are mostly perpendicular to the ground but this is not always the case, especially when accounting for the amount of noise in the 3D reconstructions, the slope differences in the fields, and the occasional but quite frequent double–planting (the seeding tractor planted more than one seeds in the same location). Consecutively, any solutions that make this assumption are doomed to fail in at least one of the aforementioned cases and cannot be employed for a generalized solution.

An observation regarding the topology of the plants that always holds however, dictates that a rain drop that falls on any part of the plant has to glide on top of the plant's surface before it reaches the ground and has only two routes to achieve that; fall over the edge of a leaf, or follow the stem closely until it reaches the plant base. The core of our algorithmic approach makes use of this observation and tries to simulate the behavior of hundreds of randomly placed rain drops, were they to glide on the surface of the point cloud P_{veg}. The associated algorithm is called Randomly Intercepted Nodes (RAIN) and records common routes of the randomly placed rain drops.

The RAIN is heavily utilized for the extraction of several of the proposed phenotypic characteristics. By changing the thresholds, altering the conditions, and even applying it iteratively, different plant topologies are captured rendering it an inseparable tool of this pipeline. In the following paragraphs we will follow the algorithm provided in Algorithm 4 and explain its basic steps.

The algorithm's name is a metaphor of a physical phenomenon, therefore when mentioning "rain drops" we simply mean randomly selected points within the point cloud. The routes of these drops while moving from point to point are simply sets of the indeces of the visited points and the selection of each next point in the path is subject to a few simple rules influenced by gravity. The goal is to populate a vector P_{labels} associated with each one of the cloud points with values that depict the *label* of each point. At the end, the points with the same *label* in the P_{labels} will belong to the same topology (stem, plant, leaf, etc.) and the points that were not visited by the algorithm ($label = 0$) and can be discarded.

Initially, we select the number of randomly generated rain drops N_{drops}. In a typical 3D reconstruction the algorithm needs to identify as many as forty plants and one thousand N_{drops} are sufficient. It makes sense for the drops to start at the higher levels

of the reconstruction to cover more ground and capture a larger portion of each plant, so any point generated lower than th_{height} is discarded and this random pick does not count towards the total N_{drops} (lines 10-12). A random drop pt with the proper starting height will be generated (line 8) and added as the first entry in a vector **Line** that stores all the visited points in the current path (line 14). The boolean variable $reachedEnd$ (line 15) determines when the current path terminates and is set $true$ when the current path reaches an already visited path or the algorithm can not find a neighbor that is lower that the current point.

Using the newly generated drop and the precomputed $kdtree$, all the neighbors within a sphere with radius th_{neigh} are found (line 17) and their height (z) is compared against the current point pt. The neighbor which is further below the pt is selected ($select$) as the next point of the path (line 18). This step can be enhanced with the computation of a derivative along the z-axis but is not necessary for the basic case. For the case of an original path that has not been visited before, the algorithm shall check if the next point is actually lower than the current point (line 24), will make the next point current (line 25), will add the next point's index in the **Line** vector, and will repeat the process from the while loop (line 16).

If the next point belongs to a path that has been visited before (line 19), all the previously stored points of the path will be combined with the old path bearing the same label, the $label$ will discard this path (line 22), and the **Line** and $reachedEnd$ will reset. In case the current point pt is the last of its path (line 27), the algorithm will verify that the path has at least $minPathSize$ points and will generate a new path with a new $label$ (line 28). In any other case, an invalid path has reached (line 30), the path variables will reset (lines 31-34) and the algorithm will start again on line 7.

The RAIN is very efficient since out of all the randomly generated drops N_{drops}, the number of original routes $N_{original}$ that need to complete the whole algorithm tend to match the number of plants N_{plants} in the reconstruction ($N_{drops} \rightarrow N_{original} \cong N_{plants}$). Most of the random drops encounter an already visited point and terminate prematurely and, given the downward exploratory movement of the drops, the number of points that are actually considered as potential path candidates are severely reduced. Therefore, the complexity of the algorithm is almost linear to the number of plants multiplied by the points in each path m and by the complexity of the $kdtree$ search ($\mathcal{O}(\log n)$, n is

the size of P_{veg}); ($\mathcal{O}(N_{plants} m \log n)$).

4.4 Stem Detection

The plant stems impose complexities in the separation of leaves, while on the other hand reveal helpful information regarding the position of leaves. Therefore, their detection is both necessary and desirable.

We call forth the skeletonization and RAIN algorithms described in Secs. 4.2.4 and 4.3.1 and set them inside a loop to iteratively act on the P_{veg}. The idea is that each iteration will be performed on a progressively pruned version of P_{veg} until only the stems survive. The pruning of the point cloud is inspired from the physical interpretation of RAIN and can be thought of as localized raining; each plant has a "private cloud" whose rain drops are falling based on a Gaussian distribution with mean right above the plant and a very narrow standard deviation. The details of the algorithm can be seen in Fig. 4.6.

The initial point cloud is treated once by the general skeletonization and RAIN algorithms to produce some initial clusters of points that may include the actual stems. The same process is then applied per cluster with the random selection of initial path points is constrained above the cluster instead of the whole point cloud. As seen in Fig. 4.7, a few iterations later, the stem point clusters have been exposed but with them there is a chance of capturing clusters of leaves. The filtering of the true stems is achieved by the detection of the corn rows and the prior knowledge that all stems need to be on a row; if a cluster of points does not belong on a cluster, it is discarded.

For the row detection we compute the lowest point of each cluster and project it on the x-y plane. In case the stem is comprised by two or mode clusters, we assume that the projections of all these lowest points will be very close to the row line. The projection of the lowest points of all the clusters on the x-y plane and the two lines that represent the two rows can be seen in the Fig. 4.8. The detection of the row lines treats these projections as 2D points on a plane and uses RANSAC to fit lines iteratively. Every time a line is found successfully, its points are removed from the search space of the algorithm until there are very few points left, or the fitting score is not acceptable. Similar to the process used for the detection of the ground plane, RANSAC is ideal

for applications with noise and uncertainty; the idea of trial and error with randomly selected data points has brought substantial results both in the case of RANSAC and RAIN.

Once the points of the stems have been found, they are immediately removed from the P_{veg} leaving behind points that belong to leaves. A filtering step similar to the one described in Sec. 4.2.3 cleans the leaves from any noisy or leftover stem points and the resulting point cloud is fed to the leaf segmentation algorithm described in the next section.

4.4.1 Limitations

Sometimes, the density of the canopy or the difference in growth between two neighboring plants can result in one plant being overshadowed. Using the proposed algorithm, if one plant is missed during one of the iterations then it can not be recovered. This is a limitation that can be addressed by setting hard thresholds on the height th_{height} of the initialization of rain drops. This step might produce more refined results but is not generalizable and each reconstruction might need different thresholds depending on the plant growth, or the slope of the ground. The example in Fig. 4.7 depicts two rows of corn of the same growth stage but one row was treated with less Nresulting in lower biomass. This would affect the threshold selection negatively.

The most common problem encountered in algorithms that are based on random initialization is the repeatability of the results. In the RAIN algorithm, this problem is manifested through the differences in the paths that are generated when the algorithm executes multiple times. Although the number of segmented plants is the same and the main shape of the stems is captured, the paths of the "rain drops" that generate these segments will be slightly different every time, thus having issues with small plant parts that one time will be part of the stem, while another time part of a leaf. The beauty of the randomization that allows for such elegance in the RAIN algorithm, at the same time is its greatest limitation.

Partially reconstructed plants provide poor results, especially the ones at the borders of the reconstruction. This creates an issue on how to select the best part of the reconstruction in order to retrieve consistent and satisfactory results. This problem can be partially addressed by the row detection methodology by informing the user which

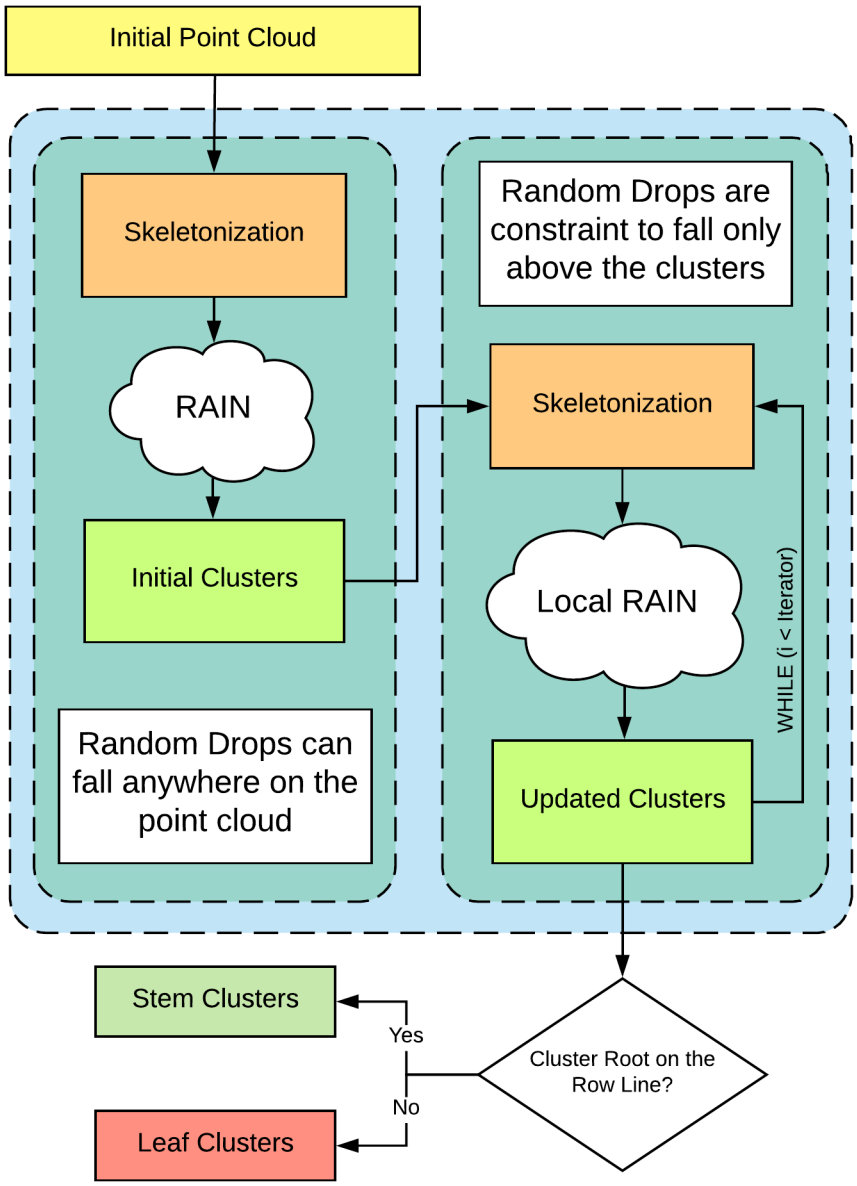


Figure 4.6: Flow chart of the algorithm for the detection of the plant stems. The initial point cloud is treated to provide initial clusters that potentially hold the stems and then an iterative process removes the non-stem points until the clusters represent either a stem or the part of a leaf. Finally, we utilize the prior knowledge of all the stems belonging on a row to throw away the leaf clusters.

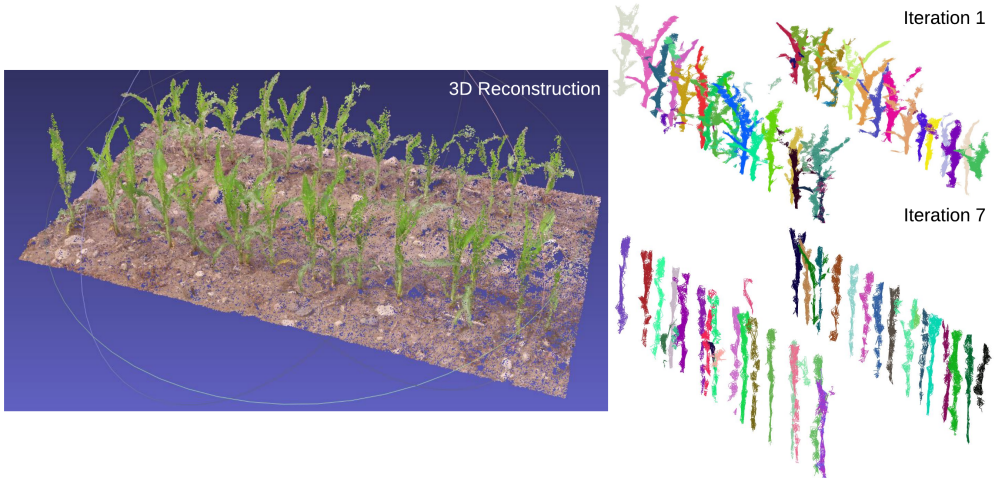


Figure 4.7: Results of the iterative algorithm for the detection of stems. The original point cloud is provided on the left for comparison. The first application of the generalized skeletonization and RAIN algorithms produces clusters with stems and leaves mixed (top right), while after the 7th iteration, only the stems are left along with some clusters that represent leaves.

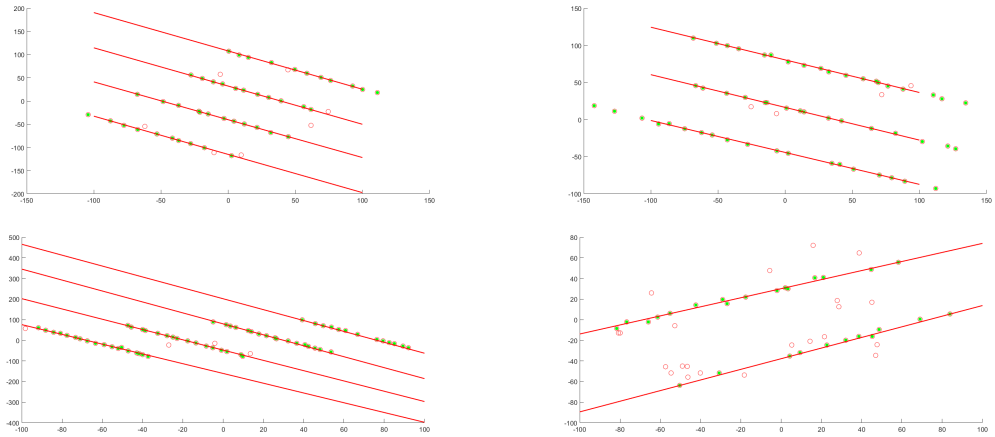


Figure 4.8: These four examples of growth stages from "V3" to "V6" demonstrate the detection of rows through the fitting of 2D lines in a collection of 2D points that were created as the projections of the lowest points of the generated individual plant clusters. The line fitting RANSAC algorithm finds the dominant lines and then terminates since the number of the remaining points and the fitting score are not satisfactory.

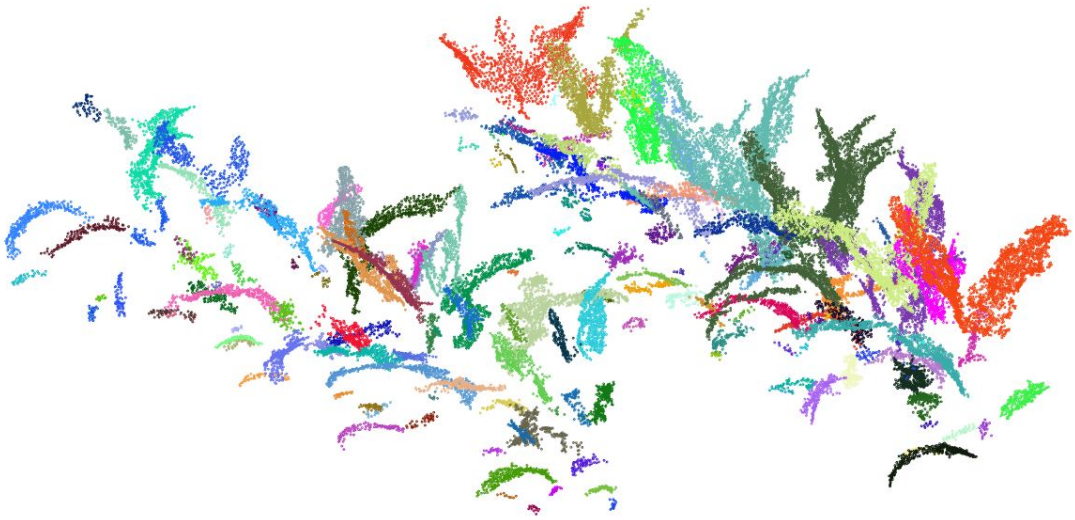


Figure 4.9: After the removal of the ground and stems using the aforementioned iterative algorithm, only clusters of leaves are left. Here, the leaf clusters have gone through a euclidean clustering step that creates groups of at least one leaf.

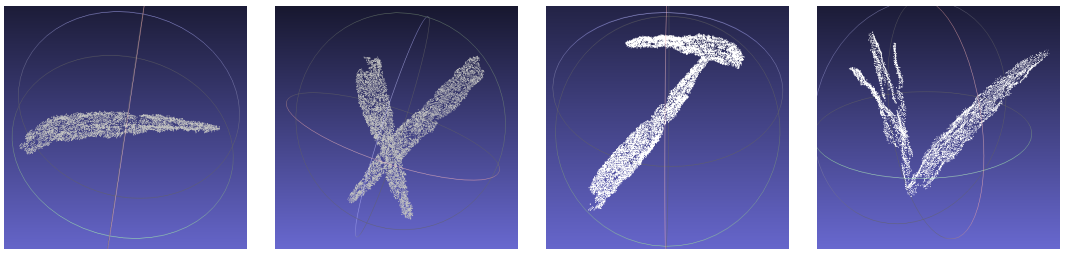
rows have the best fit. The RANSAC is going to select first the line that satisfies the points giving an indication of which row has the most and better reconstructed plants.

4.5 Leaf Segmentation

Till this point, the segmentation pipeline has managed to remove the ground, separate each plant, and extract parts of the stems leaving only the leaves as part of the initial point cloud, as seen in the example Fig. 4.9. For the growth stage of the plants we are interested in ("V3" to "V6"), the canopy is not particularly dense and most of the individual leaves are observable during the 3D reconstruction.

Still, occlusions and leaf intersections are present leading to a non-trivial segmentation process which concludes in two steps. Initially, a clustering technique such as the euclidean distance based cluster extraction combined with a statistical outlier removal [72] is employed to break down the point cloud P into N clusters of 3D points $P_i \subset P$ with $i = 1, \ldots, N$, each containing one or more single leaves, with examples seen in Fig. 4.10.

The second step of the segmentation needs to iterate over all N clusters, leave the



(a) Example of single leaf cluster.

(b) Example of two intersecting leaves in the same initial cluster.

(c) Example of two touching leaves in the same initial cluster.

(d) Example of four leaves at the top of a single plant in the same initial cluster.

Figure 4.10: Example of clusters produced by the initial euclidean clustering step. Cases like (a) do not need extra refinement since the initial cluster represents a single leaf. In the contrary, cases like (b), (c), and (d) need to be further refined to produce single leaf clusters.

single leaf clusters intact, and break apart the clusters that contain more than one leaf. For this purpose a 3D skeletonization technique is employed [73] which acts on each point cluster P_i and delivers a skeleton S_i comprised from a set of connected nodes $s \subset S_i$. Each node is the centroid of a collection of neighboring 3D points and acts as their representative.

The nodes capture the topology of the leaves, which is used by our proposed algorithm to achieve a refined segmentation. In particular, we assume only three types of nodes s; namely the endpoint nodes, the intersection nodes, and the standard nodes. Endpoints are nodes with only one immediate neighbor, while intersections have more than two immediate neighbors and standard points have exactly two.

Under this definition, we assume that any endpoint node signifies one end of a single leaf and that starting from an endpoint, a set of neighboring standard and intersection nodes in a smooth trajectory forms the midrib of the leaf. Referring to Fig. 4.11, one can see the nodes $s_1, \ldots, s_{10}$ and $s_{11}, \ldots, s_{17}$ forming the two leaf midribs while both curves start from an endpoint. Although we assume the midribs to start from an endpoint, it is possible for them to end at any node and the decision is based on the smoothness of the curve.

In order to define a criterion for the smoothness of a curve, we treat the nodes of a the skeleton as measurements of the trajectory of a physical object in 3D and employ a Kalman filter to decide whether a node belongs to the midrib or not. This segmentation

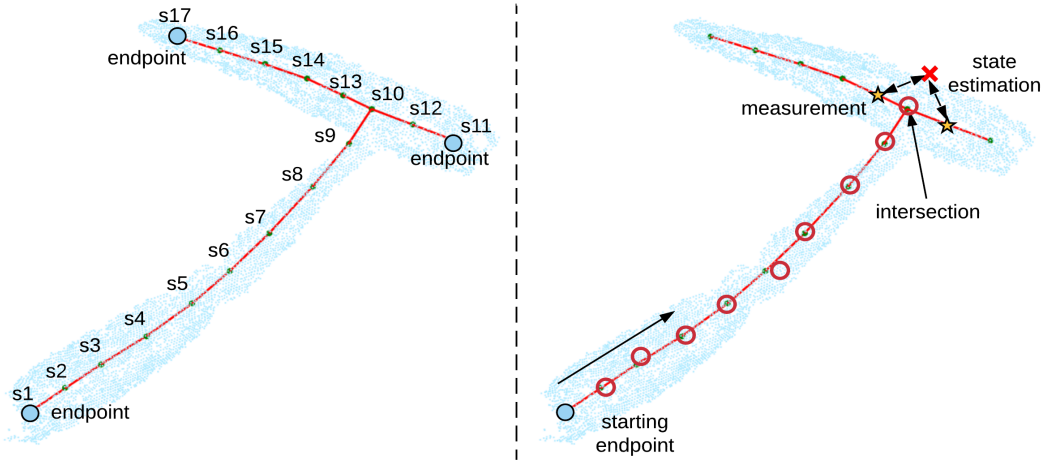


Figure 4.11: *Left:* Example of a 3D skeleton of two overlapping leaves. Three endpoint nodes (s_1, s_{11}, and s_{17}), one intersection node (s_{10}), and several standard nodes ($s_2 - s_9$, s_{12}, and $s_{13} - s_{16}$) are visible. *Right:* Example of the SKF algorithm starting from node s_1. Kalman filtering is used iteratively to make decisions on the nodes that belong to the midrib. The red circles represent estimates on the position of the nodes $x_{k_{est}}$ that have been found to be close to their respective node-measurements (green dots). The red cross presents the position estimate that is far away from its respective measurements (yellow stars), thus signifying that s_{10} is an end-node for the particular leaf.

refinement process we call Skeleton Kalman Filtering (SKF). Specifically, we employ the model:

$$x_k = x_{k-1} + v_{k-1}\Delta t + \frac{1}{2}\alpha\Delta t^2 \tag{4.6}$$
$$v_k = v_{k-1} + \alpha\Delta t$$

which is transformed, as used in Kalman filtering, in the matrix form:

$$\begin{pmatrix} x_k \\ v_k \end{pmatrix} = \begin{pmatrix} I_3 & I_3\Delta t \\ 0_3 & I_3 \end{pmatrix} x_{k-1} + \begin{pmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{pmatrix} \alpha \tag{4.7}$$

with $x_k = [x, y, z]^T$, $v_k = [\dot{x}, \dot{y}, \dot{z}]^T$, $\Delta t = 1$, $\alpha = 0.01$, $I_3 \in R^{3x3}$ identity matrix, and $0_3 \in R^{3x3}$ zero matrix. At the same time, the measurements $z_k = [z_{k_x}, z_{k_y}, z_{k_z}]^T$ take the form:

$$z_k = \Big(I_3|0_3\Big)\, x_{k-1}, \tag{4.8}$$

and the model and measurement uncertainty matrices Q and R respectively are:

$$Q = \begin{pmatrix} I_3\frac{\Delta t^4}{4} & I_3\frac{\Delta t^3}{2} \\ I_3\frac{\Delta t^3}{2} & I_3\Delta t^2 \end{pmatrix},\ R = \Big(I_3\sigma^2\Big), \tag{4.9}$$

with $\sigma = 0.1$.

The position part x_k of the state vector is initialized with the first endpoint of the midrib, velocity $v_k = [0, 0, 0]^T$ and the Kalman filter is applied iteratively. At each iteration the estimated position $x_{k_{est}}$ is compared with the next neighboring node(s) (that act as measurements z_k) and based on a distance threshold the node is considered part or the end of the midrib.

$$||x_{k_{est}} - z_k||_2 < d_{threshold}. \tag{4.10}$$

As seen in the right image of Fig. 4.11, the skeleton branch that initiates from the starting endpoint s_1 applies Kalman filtering and adds the standard points up to s_{10} to the midrib. The filter estimates that the next node s_{11} should be at the red cross for the smoothness of the midrib to persist. Nevertheless, the actual neighboring nodes (yellow stars) are not consistent with the estimate meaning that the intersection node s_{10} signifies the end of the leaf.

4.5.1 Limitations

The proposed methodology depends heavily on the quality of the 3D reconstruction for the segmentation of individual leaves. Especially in later growth stages, the density of the canopy occludes heavily the lower leaves resulting in their partial reconstruction. This in turn affects significantly the number of leaves that are estimated and therefore poses problems later in the extraction of phenotypic characteristics.

A partial 3D reconstruction also affects the SKF segmentation step which utilizes the 3D skeletonization to separate the independent leaves. When the leaf surface has a large hole due to lack of texture in the reconstruction process, the skeleton is forced to create unnecessary branches which may end up in the over-segmentation of the leaf. This problem is partially addressed by the flexibility of the SOM and the summation of all the final leaf areas, nevertheless, it introduces inaccuracies to the final area computation.

Finally, a limitation inherited from the Kalman filter is observed when the meeting angle between two leaves is sharply acute. During the tracking of the s_i points on a leaf, at the point of intersection the tracker may select to continue with the wrong leaf. This might not affect any computations regarding the leaf area or the leaf number, but is still a limitation to be addressed.

4.6 3D Datasets

The several developed algorithms were executed on both artificial and real corn plants. Due to the complexity of the problem, the reproducibility of the experiments, and the accuracy in the collection of the ground truth measurements we decided to validate the numerical correctness of some complex biometrics on artificial corn stalks, while the real corn reconstructions were limited in the span of “V3” and “V6” growth stages. Dense

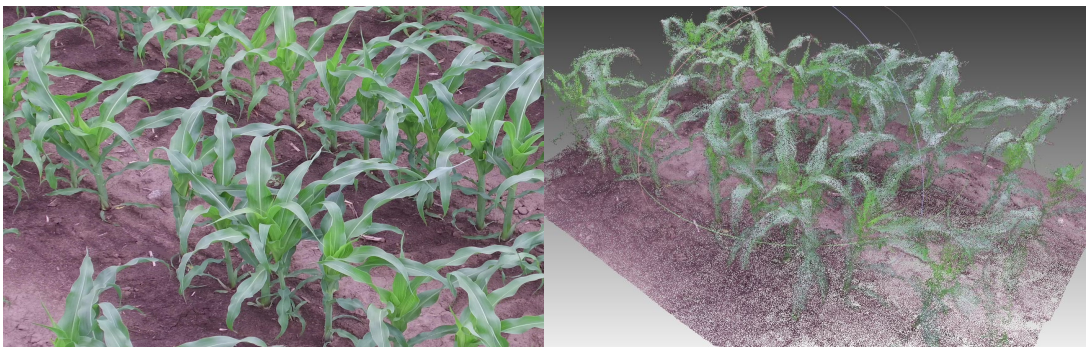


Figure 4.12: *Left:* One of the images used for the construction of the 3D model visible on the right. *Right:* The dense 3D reconstruction of real corn plants in the field is provided as reference for visual comparison against its RGB image.

canopies with heavy occlusions in the lower leaves were dismissed as inapplicable to the developed algorithms due to the limitations mentioned in the previous sections.

4.6.1 Artificial Corn Data

We used a total of six artificial plants with similar biometrics that are based on real corn models at a "V6" growth stage. The basic biometrics (height, leaf length, inter–nodal distance) were measured directly. The areas of their leaves was approximated by the formula: $L * W * k$, with L the length of the leaf from the stem to the tip, W the maximum width, and the constant $k = 0.75$ was selected based on literature as a viable approximation of the leaf area [74].

The 3D reconstructions were created offline using the VisualSFM toolbox [69] with the number of input images varying from 18 to 24 and the images were collected with a handheld Olympus TG-4 camera of 1440x1920 pixel resolution. A sample sparse reconstruction output from the VisualSFM toolbox can be seen in Fig. 4.3, while dense reconstruction results produced via the use of the PMVS tool [75] are seen later in the Fig. 5.5b.

We are considering six different configurations of artificial corn stalks with increasing complexity which, as seen in Fig. 4.13, try to mimic realistic scenarios and assist in verifying the sensibility of the proposed method. In experiment #1 a single plant is reconstructed showing how the algorithm behaves with minimal occlusion. The same

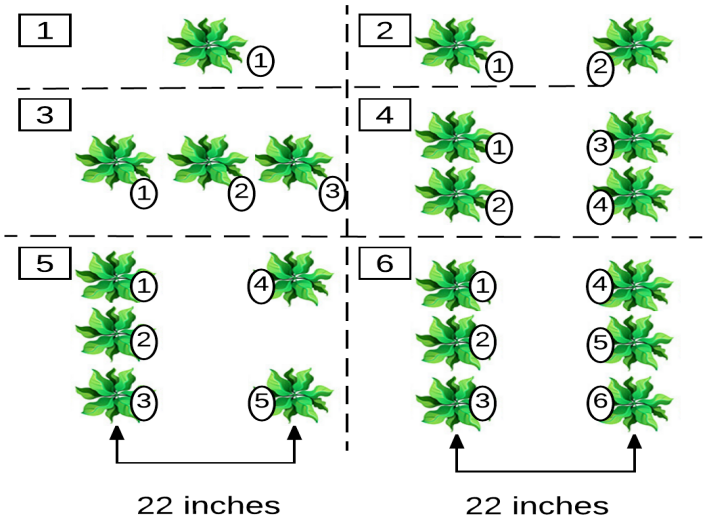


Figure 4.13: The experimental setup involved six different corn plant configurations with increasing complexity. The numbers inside the squares depict the experiment number and the numbers in the circles represent the plant identification. One to six plants were used in realistic scenarios keeping distances between rows at 22 inches, a standard corn row distance used at the United States farmlands.

goes for experiment #2 where non-overlapping leaves are considered in a standard 22 inch distance between two corn rows. Experiments #3 through #6 show cases of severe overlap and occlusion with increasing number of plants and #5 shows a case where the row is missing a plant due to seeding error.

4.6.2 Real Corn Data

We collected 15 video sequences from three different field locations in Minnesota (Becker, St. Paul, and Rosemount Fig. 3.6), with corn stages ranging from "V3" to "V8" and row distances either 22 or 30 inches. The sequences were collected by a DJI Matrice 100 UAV with a DJI Z3 camera and span 30 seconds each. The 3D reconstructions were created offline using again the VisualSFM toolbox [69] with the input images retrieved as frames from the video sequences. Between 80 and 120 images of 3840x2160 pixel resolution were used to create each reconstruction. Image samples and their 3D reconstructions can be seen in Fig. 4.12. Due to inconsistencies in results for the "V7" and "V8" stages, these videos were discarded.

4.7 Practical Findings

Through the presentation of the algorithmic point cloud processing and the examination of its limitations it was made clear that the visual similarity between original photos and the reconstructed canopy is of great importance. Although the process is designed to be agnostic to the means of image collection, this step needs to be carefully examined as the more detailed the 3D model is, the more accurate the phenotype extraction will be. The use of a UAV moving in a circular trajectory aims to capture as many different viewpoints as possible and works well for real field cases since it can be automated using the manufacturer's software interface.

In a realistic scenario, a group of about forty well reconstructed corn plants can be covered in less than four minutes meaning that this application is targeting the subsampling of a field instead of a complete coverage. Nevertheless, a typical twenty minute UAV flight can produce a plethora of measurements much faster and than a human and the resulting 3D reconstructions can be saved for later use as archives and references. An interactive 3D visualization tool integrated with the proposed algorithmic processes can be used for measurement extraction and performance verification from the comfort of ones office.

In the next chapter, the several extracted plant characteristics exemplify how the algorithms presented above can lead to better phenotyping of the crop canopy. Characteristics that were previously measured sparsely and without any means of verification, now can be automatically estimated, extracted for several plants at a time and be available for future use.

Algorithm 4: The RAIN algorithm is core for the extraction of several phenotypic characteristics.

Result: P_{labels} = vector with the labels of each visited point

Initialization:
P_{veg} – input point cloud
$kdtree = \textbf{createKDtree}(P_{veg})$
$counter = 1$
$label = 0$
Main Loop:
while $counter < N_{drops}$ **do**
  $ptIndex = \textbf{randomlySelectPoint}(P_{veg})$
  $pt = P_{veg}(Index)$
  if $pt.z < th_{height}$ **then**
    continue
  end
  $label++$
  Line $= ptIndex$
  $reachedEnd = False$
  while $!reachedEnd$ **do**
    neighborsIndex $= \textbf{findNeighbors}(kdtree, pt, th_{neigh})$
    $[d, select] = \textbf{max}(pt.z - P_{veg}(neighborsIndex).z)$
    if $P_{labels}(neighborsIndex(select))! = 0$ **then**
      $P_{labels}(\textbf{Line}) = P_{labels}(neighborsIndex(select))$
      Line $= \varnothing$
      $label--$
      $reachedEnd = True$
    else if $d > 0$ **then**
      $pt = P_{veg}(neighborsIndex(select))$
      Line $=$ **Line** $\cup\, neighborsIndex(select)$
    else if ***size***$(\textbf{Line}) >= minPathSize)$ **then**
      $P_{labels}(\textbf{Line}) = label$
      $reachedEnd = True$
    else
      Line $= \varnothing$
      $reachedEnd = True$
      $label--$
      $counter--$
    end
  end
  $counter++$
end

Chapter 5

Phenotype Extraction

The processing of the 3D point cloud as described previously in Chapter 4 was the mandatory step that generated the building blocks which will be used in this chapter to estimate the various plant phenotypic information. The selected phenotypes are examples of some commonly measured variables that are useful to both farmers and researchers alike.

The next few sections present the methodologies for the estimation of (i) the plant number in a given 3D reconstruction, (ii) the leaf area index, (iii) the individual plant height, (iv) the leaf length, and two characteristics related to the geometry of the leaves with respect to the stem; (v) the leaf angle with respect to the plant's stem, and (vi) the inter–nodal distance for a single plant.

5.1 Plant Counting

Counting the leaves in a given 3D reconstruction is the first step in the process of extracting phenotypic characteristics of corn. Successfully detecting a plant reveals information on the quality of its reconstruction and by filtering out badly reconstructed plants we can increase the success rate of the consecutive phenotypes.

Performing the preprocessing steps and the RAIN algorithm described in Secs. 4.2 and 4.3.1 on a point cloud allows the segmentation of individual plants out of a group of interacting plants as seen in Fig. 4.1.

5.1.1 Experimental Results

The results of 8 different real corn crop sequences of growth stages between "V3" and "V6", processed 20 times each, are presented in Table 5.1. The table provides the mean and standard deviation of the estimated number of plants, the groundtruth number of plants, the number of rows for each sequence, the error between the mean and the groundtruth, and the maximum error that was reached. The mean error ($error_{mean}$) is just:

$$error_{mean} = \frac{abs(true - est_{mean})}{true}, \tag{5.1}$$

while the maximum error ($error_{max}$) is defined as the percentage of the largest deviation of the estimated count from the groundtruth count:

$$error_{max} = \frac{max(abs(true - est)}{true}. \tag{5.2}$$

For example, the maximum error for the "V3 St. Paul seq.1" is (40 - (38.8 - 2.49)) / 40 = 9.21%. This shows that the pipeline is adaptive to the number of rows and can handle a large range of reconstructed plants (from 16 and 1 row to 64 and 4 rows).

The standard deviation from the mean value is a result of the random character of the RAIN algorithm and the RANSAC process for the detection of the rows; the "rain drop" paths are not the same for each execution, and the 2D lines that determine the rows are slightly different each time. The deviation increases as the 3D reconstruction quality decreases with the maximum being observed for "V6" where the 3D reconstructed stems and lower leaves are occluded by the dense canopy, therefore providing a less accurate 3D model. Furthermore, in "V6" the upper leaves are long and curved and their curvature forces several of the "rain drops" towards their tip. This means that the row detection RANSAC algorithm deals with higher uncertainty when fitting the 2D lines as seen in the lower right example of Fig. 4.8 and it misses some of the actual plants.

5.2 Leaf Area Index

In the agriculture literature a common measure that indicates the biomass of the plant is the Leaf Area Index (LAI). This dimensionless quantity is defined as the one-sided

Sequence	# Plants E.	# Plants GT.	# Rows	$error_{mean}$	$error_{max}$
V3 St.P. seq.1	38.8 ± 2.49	40	2	3.00%	9.21%
V3 St.P. seq.2	53.6 ± 2.01	54	4	0.74%	4.47%
V4 B. seq.1	14.1 ± 0.32	16	1	11.88%	13.85%
V4 B. seq.2	52.9 ± 0.99	51	3	3.73%	5.68%
V5 R. seq.1	45.3 ± 1.80	46	3	1.52%	5.44%
V5 R. seq.2	63.2 ± 1.62	64	4	1.25%	3.78%
V6 St.P. seq.1	22.4 ± 6.98	23	2	2.61%	32.95%
V6 St.P. seq.2	33.4 ± 9.43	36	2	7.22%	33.42%

Table 5.1: This table presents the mean and standard deviation of the estimated number of plants in different sequences (# Plants E.) for growth stages between "V3" and "V6". There is also the groundtruth (# Plants GT.), the number of rows (# Rows), the error between the groundtruth and the estimated mean ($error_{mean}$), and the maximum error observed ($error_{max}$) for each experiment. Each sequence was run 20 times to verify the repeatability of the RAIN algorithm. The abbreviations in the sequences names reveal the location the data were collected from; St.P.: St. Paul, MN, B.: Becker, MN, R.: Rosemount, MN.

green leaf area per unit ground surface area [76]. This rather generic definition has received several practical definitions depending on the plant species, the leaf shape and the particular application [77].

For broadleaf plants such as corn, LAI can be computed either directly by destructive sampling of canopy leaves, or indirectly by approximate techniques that involve 2D imaging and solar radiation measurements above and below the canopy. Chen et al. [78] and Bréda [79] are presenting and comparing a variety of methods of both direct and indirect LAI estimation techniques. Direct methodologies produce accurate results but are time consuming and destroy the plants. On the other hand, estimating the LAI based on the existing indirect methodologies requires human presence for the collection of data [80], and may result in estimation errors of up to 25% because of occlusions and cluttering introduced by the dense canopies [81].

Through remote sensing, promising indirect approaches for the measurements of the spatial variability in LAI have been proposed [82], with their application in real world scenarios being infrequent and spatially limited. Via the use of detailed 3D models of individual crops, we propose a methodology to alleviate these drawbacks and provide LAI measurements that can be used for daily updates of crop growth models, and enhance the ability to estimate crop nutrient requirements.

A popular interpretation of the LAI when dealing with canopy imaging considers

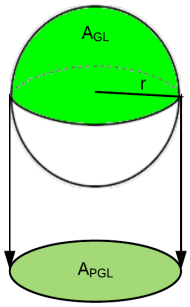


Figure 5.1: A simple example to explain the definition of LAI. The total area of the green part of the sphere symbolizes the area of the green leaves and is $2\pi r^2$ (r is the radius), while the orthogonally projected area is a circle of area πr^2. Therefore, LAI $= 2\pi r^2/\pi r^2 = 2$.

the ratio of the total area of the green leaves to the area of the green leaves when they are projected on the ground:

$$LAI = A_{GL}/A_{PGL}, \tag{5.3}$$

with A_{GL} denoting the **A**rea of the **G**reen **L**eaves and A_{PGL} the **A**rea of the **P**rojected **G**reen **L**eaves. This interpretation of the LAI index receives values ≥ 1 since A_{PGL} is generally smaller than A_{GL} considering the various occlusions amongst leaves [80].

In order to estimate the LAI based on its original definition as provided by Eq. 5.3, we need to compute the two quantities that correspond to i) the one-sided green leaf area (A_{GL}) and ii) the unit ground surface area (A_{PGL}).

Even with the refinement step of SKF 4.5, significant problems manifest with the creation of a 3D point cloud due to noisy measurements and textureless leaf areas. In particular, noise corrupted points that were originally part of the actual leaf surface are reconstructed away from it and are hard to detect and correct. Furthermore, lack of texture generates large uneven holes in the central parts of the leaves that require supervised hole filling algorithms [83] in order to create a meaningful surface for the area computation.

A computationally efficient algorithm that produces satisfactory results overcoming the aforementioned limitations is the Self-Organized Map (SOM) [84]. Utilizing two fully

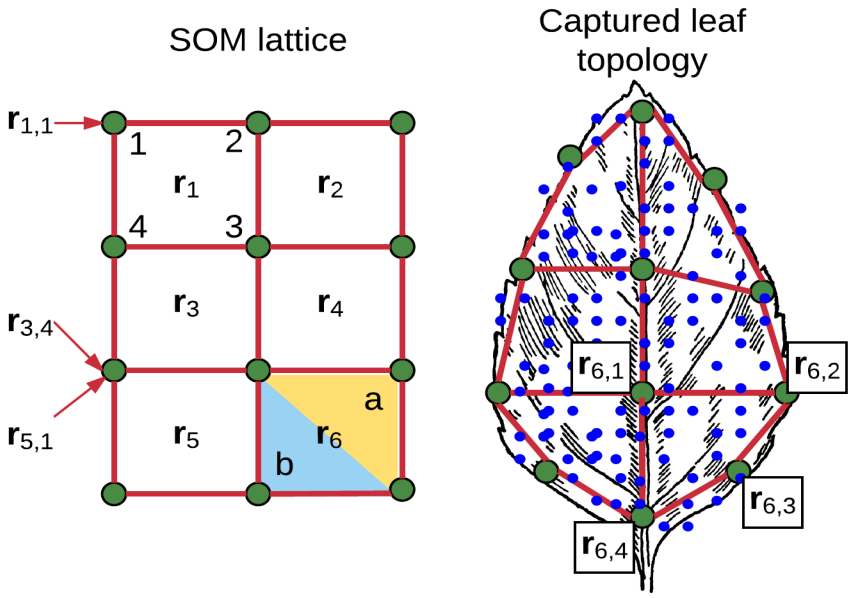
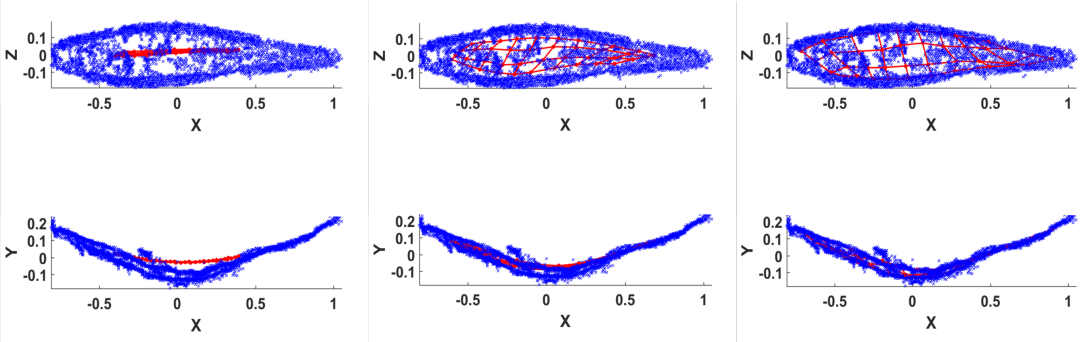


Figure 5.2: The Self-Organized-Map (SOM) lattice (left) is adapting to the surface of the leaf (right) capturing the topology of the reconstructed points. *Left:* Each square in the lattice receives an identification label r_u and its four vertices (larger dots) are numbered in a clockwise fashion starting from the top left $v = 1, 2, 3$, or 4. This way, the computation of the area through the two triangles (a and b) is feasible. *Right:* The lattice assumes the reconstructed points' (small dots) topology. On a 3D surface, the SOM lattice resembles a membrane that stretches around the 3D points. An example of the sixth polygon's indeces is visible inside the squares.



(a) $t = 200$ (b) $t = 1000$ (c) $t = 6000$

Figure 5.3: Progression of the Self-Organized Map. The initial lattice (red grid) is iteratively trying to capture the topology of the leaf (blue dots represent the 3D points of the leaf). The outliers are ignored and the resulting surface is resembling the real leaf despite the reconstruction errors. Top row is a top-down view of the leaf, bottom row is a lateral view. t symbolizes the time step in the algorithmic process. In MatLab the average processing time of one leaf is 0.52 seconds.

connected layers of a neural network, SOM is an unsupervised algorithm which creates a grid that organizes itself to capture the topology of the provided data. The SOM is undertaking an automatic smoothing of data by fitting this grid of points according to the density of the recorded data, thus minimizing fitting error in densely reconstructed areas. The grid, also known as lattice, provides flexibility in that its granularity is controlled at will by the user. Altering this parameter can achieve higher execution speed so that the estimation of the LAI is performed without affecting the accuracy to undesirable levels.

In our case, the provided data is a point cloud of a leaf and the SOM is used to express the surface of the leaf so that the area computation is feasible. When the SOM converges to its final form (Fig. 5.3c), several four-edged polygons with known vertices cover the surface of the leaf and are identified as $r_{u,v}$, with $u = 1, \ldots, K$ number of polygons and $v = 1, 2, 3,$ or 4 the number of the vertex in a clockwise order (Fig. 5.2). Each polygon is then broken down into two triangles by indexing its four vertices v in order $1 \rightarrow 2 \rightarrow 3$ (triangle a) and $3 \rightarrow 4 \rightarrow 1$ (triangle b). The total area of the surface can be approximated by adding the areas of all the triangles. Finally, the total green leaves area (A_{GL}) is the summation of the areas of the individual leaf clusters.

A_{PGL} can be computed in a simpler manner. First, all the vertices of the SOM lattice are treated as 3D points and are projected on the ground with the projection

matrix:

$$T = I_3 - nn^T \in \mathbb{R}^{3x3}, \tag{5.4}$$

with $I_3 \in \mathbb{R}^{3x3}$ the identity matrix and $n \in \mathbb{R}^3$ the normal vector of the ground plane. The projected points form a concave two dimensional polygon in 3D which is used to create a mesh of triangles that occupy the desired area. An occupancy grid with known cell size is generated and overlayed ontop of the mesh to determine how many cells are occupied. Again, adding the area of all the occupied cells provides an estimation of A_{PGL}.

5.2.1 Experimental Results

Results on the algorithm that estimates the area of the leaves is present in Figure 5.3. The figure shows the expansion of the network over several iterations until it converges. Two viewpoints are provided to clarify the progress steps. The SOM algorithm has proven to be particularly robust and manages to adapt to the data providing a leaf-like shape and overcoming limitations such as noise 5.6(a), small number of points 5.6(b) and sparse reconstruction 5.6(c).

The validity of the algorithm is assessed by comparing the computed A_{GL} scaled using Eq. 4.2, against the groundtruthed total leaf area (A_{GL}) measurements of the artificial dataset. In Table 5.2, each column represents one of the six experiments and the rows hold the estimated leaf area information for each plant. The total leaf area is the sum of the individual plants' leaf area and the accuracy of the methodology is validated by the relative error which is computed as:

$$error = \frac{abs(T - GT)}{GT}. \tag{5.5}$$

The estimated LAI is presented last for completeness despite the lack of means to compute the groundtruthed projected leaf area (A_{PGL}).

Further, in Fig. 5.4 we present some examples of single leaf area estimation taken

		No. of Experiment					
		1	2	3	4	5	6
Corn Stalk ID	a	0.1198	0.1062	0.1205	0.1392	0.1328	0.1216
	b	-	0.1241	0.1137	0.1361	0.1294	0.1090
	c	-	-	0.1251	0.1103	0.1405	0.1377
	d	-	-	-	0.0989	0.1002	0.1150
	e	-	-	-	-	0.0975	0.1281
	f	-	-	-	-	-	0.1000
	T	0.1198	0.2303	0.3593	0.4845	0.6004	0.7114
	GT	0.1119	0.2238	0.3357	0.4476	0.5595	0.6714
	error	6.59%	2.82%	6.57%	7.62%	6.81%	5.62%
	LAI	1.6954	1.6667	2.6882	2.6994	1.8296	2.7205

Table 5.2: Leaf area estimation for each one of the plants in all six experimental setups. The mean ground truth total leaf area for a single artificial plant is $A_{GL} = 0.1119$. Since all six artificial plants are industrially manufactured, we assume they share approximately the same total leaf area. The variable T represents the estimated total leaf area, which is a column-wise sum of each experiment and GT is the groundtruth total area of the reconstructed plants. The LAI is computed for each experiment and presented in the last row.

from the most complex experiment #6. An image of the leaf is placed next to its 3D reconstruction followed by the estimated and groundtruthed values of its area presented in Table 5.3. These results support our proposed methodology and suggest that we may aim for a generalizable version that can be utilized as a tool for the estimation of LAI and the replacement of outdated and labor intensive solutions.

Leaf #	1	2	3	4	5
Area	0.0204	0.0062	0.0122	0.0031	0.0320
GT	0.0168	0.0055	0.0051	0.0086	0.0261
Leaf #	6	7	8	9	10
Area	0.0228	0.0210	0.0190	0.0189	0.0295
GT	0.0247	0.0247	0.0165	0.0165	0.0261

Table 5.3: This table accumulates the area estimation and groundtruth (GT) of the leaf instances seen in Fig. 5.4. An interesting observation regarding the leaves #3 and #4 can be made regarding the accuracy of the area estimation. These leaves are too close for the 3D skeletonization algorithm to separate them correctly resulting to one of the two leaves dominating over the other. The summation of their respective estimated areas closely follows the groundtruth.

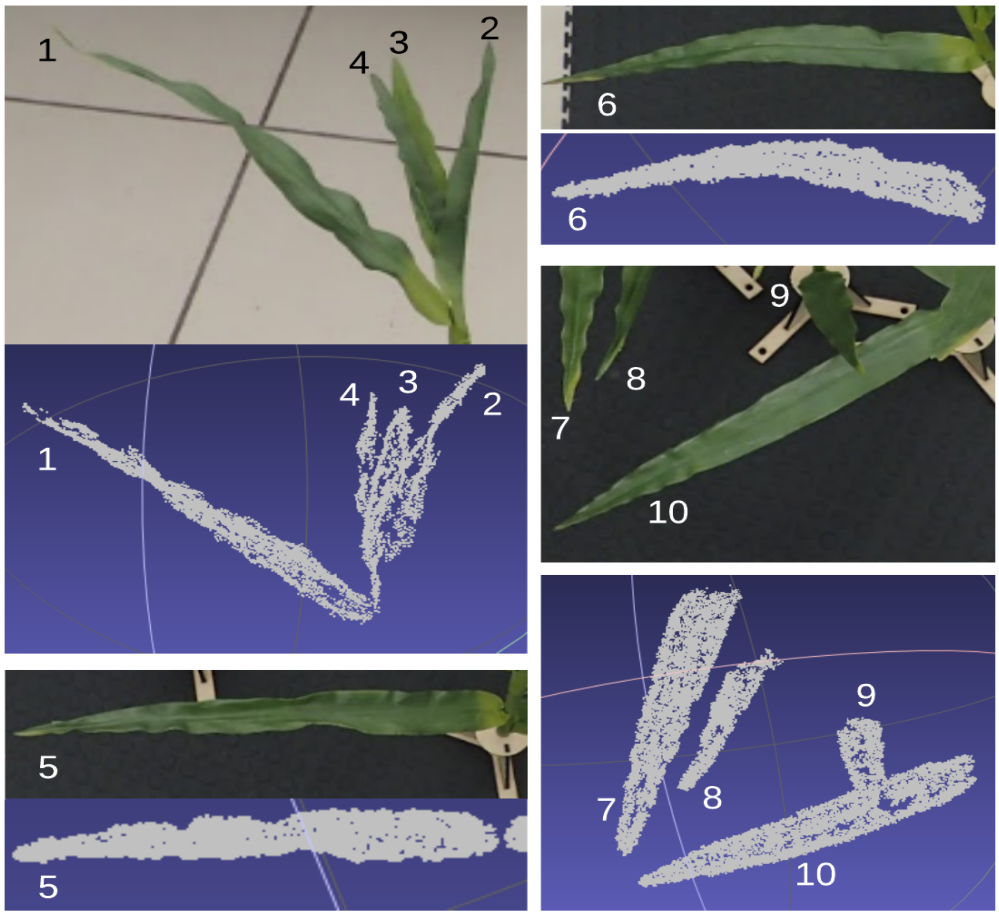
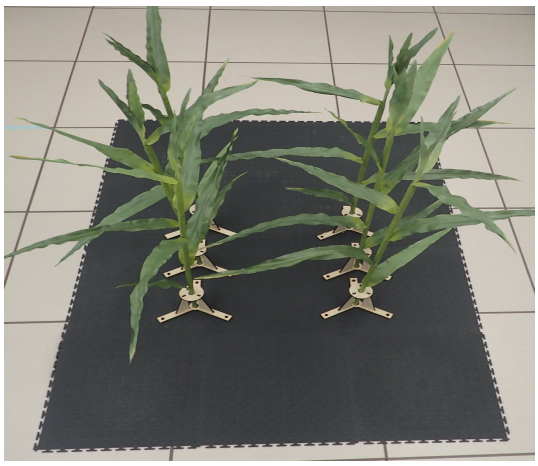


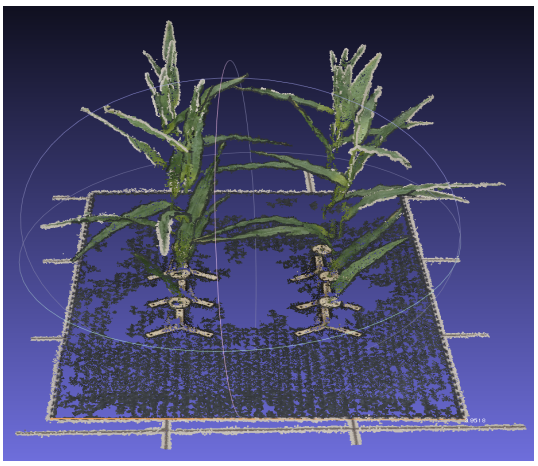
Figure 5.4: Several examples of initial images along with their reconstructed pairs are presented. In Table 5.3 We provide the estimated area for each leaf along with its groundtruth (GT) value.

5.3 Individual Plant Height

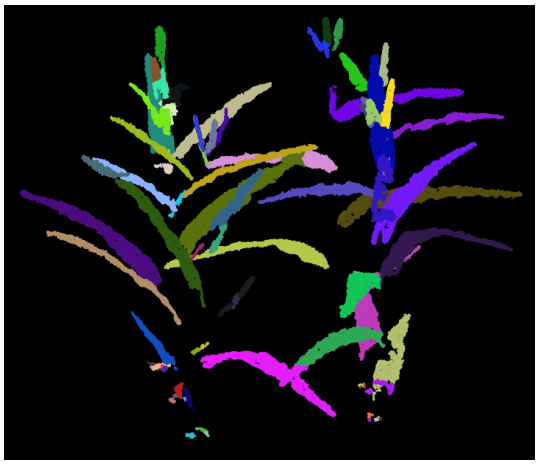
We define the height of a corn plant as the euclidean distance from the highest point of the newly emerging leaves to the lowest point that touches the ground. This definition was chosen to avoid inaccuracies arising from plants with stems not perpendicular to the ground and the unpredictable changes of the ground elevation. The distance of the highest plant point to its projection on the ground may be prone to error if the plant is not completely perpendicular to the ground. From a visual inspection of Fig. 5.8 it is apparent that the newly emerging leaves are growing above the plant acting as an extension of the stem. Since the positioning of those new leaves depend on the crop



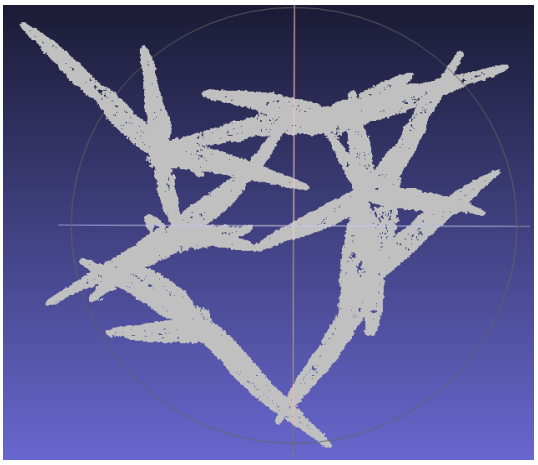
(a) Lateral view of six artificial corn stalks.



(b) Reconstruction of six artificial corn stalks.



(c) The clusters created after the euclidean distance based segmentation and the SKF based refinement step. Each color is a different cluster.



(d) The projection of the reconstructed points on the ground plane. The overlap between the leaves is apparent and represents the occlusions among the leaves.

Figure 5.5: Steps for the point cloud preprocessing. Several images similar to (a) create a 3D point cloud (b). The segmented leaves (c) as well as the projection (d) are used to compute the nominator (A_{GL}) and denominator (A_{PGL}) of the LAI.

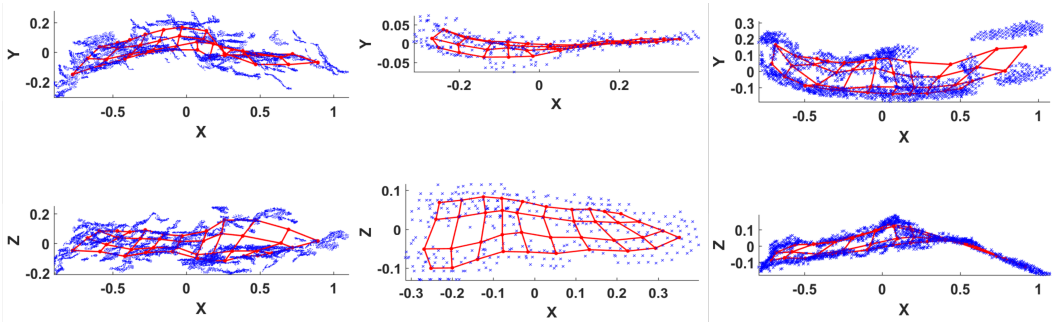


Figure 5.6: Extreme cases of reconstructed leaves validate that the algorithm performs satisfactorily. In (a), a small number of images was used for the reconstruction, resulting in a noisy point cloud. The reconstructed leaf in (b) is small and further away from the camera, therefore its 3D points are fewer and sparse. A significant part of the leaf has not been reconstructed in (c).

Sequence	Height E.	Height GT.	$error_{mean}$
V3 St.P. seq.1	23.29 ± 1.01	23.19 ± 3.99	0.43%
V3 St.P. seq.2	23.38 ± 2.15	24.81 ± 3.42	5.76%
V4 B. seq.1	31.00 ± 4.12	30.64 ± 3.12	1.17%
V4 B. seq.2	41.40 ± 6.16	38.25 ± 9.70	8.24%
V5 R. seq.1	45.01 ± 6.34	40.74 ± 5.38	10.48%
V5 R. seq.2	39.10 ± 4.21	35.28 ± 4.87	10.83%
V6 St.P. seq.1	58.91 ± 4.11	54.95 ± 6.31	7.21%
V6 St.P. seq.2	58.86 ± 8.47	54.59 ± 8.52	7.82%

Table 5.4: This table presents the mean, standard deviation, and mean error percentage of the height estimation and height groundtruth of four growth stages. This is the value representation of the Fig. 5.7.

hybrid, it is helpful to include them in the plant height estimation to provide a better decision tool for the hybrid developers.

A variance of the stem segmentation algorithm 4.4 is utilized for the estimation of individual plant height. The differences are focused on two parameters; fewer iterations of the overall algorithm, and a larger standard deviation for the constrained local RAIN step. This allows the plants to keep their top leaves during the pruning steps as these are located closely above the cluster centers. The resulting clusters of plant points consist of the stem and the top leaves and the highest point of each plant cluster is obtained by a search for the maximum z coordinate.

The lowest point of a stem cluster is not always touching the ground, and this is a

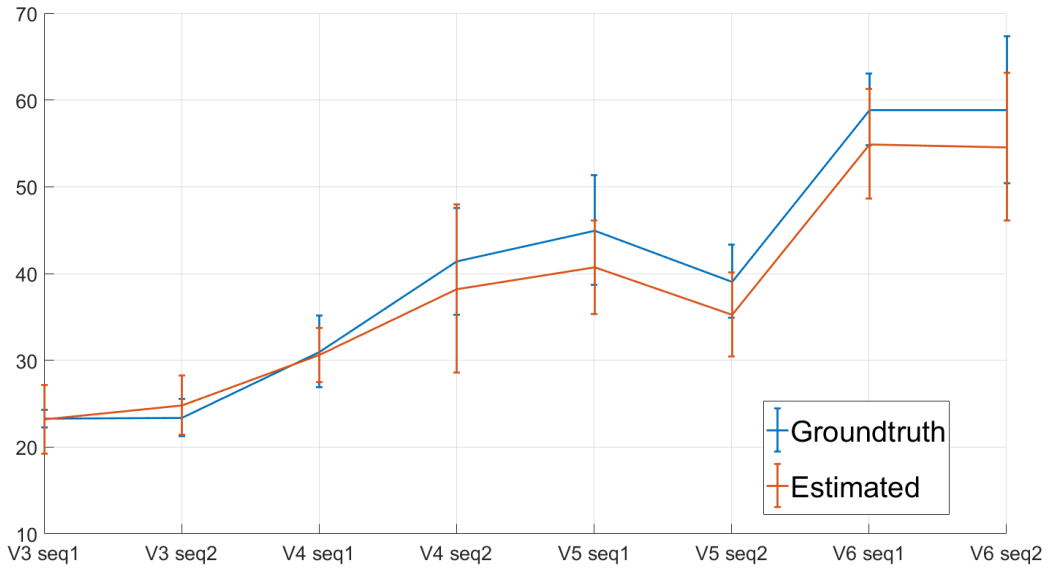


Figure 5.7: The estimated and groundtruth height of 8 groups of plants from 4 different growth stages. "V4 seq1" and "V5 seq2" represent nitrogen deficient plots and the reduction in height is apparent. The mean values are following the red and red lines while the standard deviation values are represented by the error bars.

result of a poor reconstruction. The decision on the actual lowest point is made based on a voting scheme of neighboring stems. The plants that belong to the same row vote to decide on an average lowest point and the plants that are close enough to that average are allowed to use their own lowest point, whereas the plants that are far away from it are forced to use the average. The row detection described in Sec. 4.4 is utilized for the selection of the lowest points of the plants and the height is just a subtraction between the z values of the highest and lowest points.

5.3.1 Experimental Results

A numerical comparison between the estimation and the groundtruth of the mean and standard deviation of plant heights in different growth stages, visible in Table 5.4, shows that the proposed methodology captures the trend of the plants growing as their growth stages advance. In the later stages there is a constant difference of 3-4cm and this offset is traced back to the difference in ground elevation while groundtruthing;

the groundtruth was measured as the distance from the highest point to its projection to the ground while the estimation is performed as the height difference between the highest point and the root. In most cases the highest point belongs to a leaf that is overhanging towards the middle space between two consecutive rows and the ground in between rows is dug deeper than the ground next to the root. Since the smaller plants have emerging leaves close to the center of the plant, this phenomenon is not prevalent in stage "V3" or the nitrogen deficient "V4 seq1" sequence.

The diagram of Fig. 5.7 reveals that the standard deviation of the estimated values (in red) slightly underestimates the height of the later stages because of inconsistencies in the 3D reconstruction; some of the plants, being at the border of the reconstruction or being heavily entangled with neighboring plants, show missing parts (leaves and stem) that create erroneous height estimations. The voting scheme that decides on a lowest point representative manages to reduce the amount of error but when top leaves are lost very little information can be retrieved.

The sequences "V4 seq1" and "V5 seq2" depict crops that were artificially deprived in nitrogen fertilizer and their malnourishment is visually detectable by the drop in the corn height. The proposed tool can be utilized to draw a qualitative impression of the height in different parts of the field and this can be enough for a farmer to detect deficiencies and anomalies. At the same time, interactive visual results such as Fig. 5.8 that show the plant segmentation and height estimation of real corn plants at "V5" can provide a better understanding of the dynamics between plants.

5.4 Leaf Length

The definition of distance used for the leaf length abides to the geodesic properties where the distance between two points follows the surface on which the points belong. There is a significant difference in curvature amongst leaves and just using the euclidean distance in 3D space between two points would produce inconsistent results.

Assuming that the SKF segmentation presented in Sec. 4.5 has produced clusters of points representing individual leaves, the determination of a leaf length is utilizing its elongated geometry. More specifically, the singular value decomposition of the covariance matrix created from the coordinates of the leaf points can provide the orientation

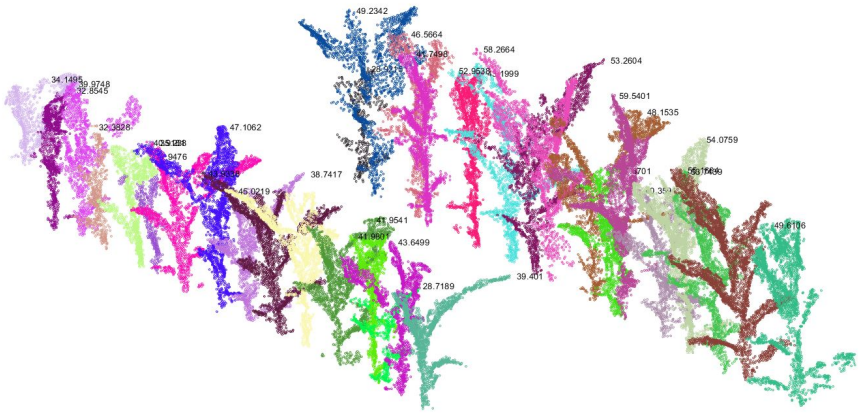


Figure 5.8: The results of the automated height computation for individual plants are seen next to the highest point of each plant.

of the elongation via the eigenvector associated with the largest eigenvalue. This vector is used to rotate the leaf perpendicular to the x-y plane using the concepts introduced in Sec. 4.2.2 (covariance matrix Eq. 4.4 and Rodrigues formula Eq. 4.5). A steepest descent algorithm follows starting from the point with the highest z and terminating at the point with the lowest z. Some results that will assist the visualization of the algorithm are visible in Fig. 5.9. The vectorial addition of all the points along the descent provides an estimate of the geodesic leaf length.

5.4.1 Experimental Results

The validation of the leaf length algorithm was completed with the artificial dataset where all 6 plants and their 6 emerged leaves were estimated. The results are comprehensively depicted in Fig. 5.10 where the mean and standard deviation of the length for each leaf (numbers 1 to 6) is compared against the groundtruth value. The same results are depicted in a different representation in Table 5.5 along with the estimation error that varies from 5.71% to 25.2%. The leaves of only one plant were used for groundtruth since all the artificial plants were identical, while the six leaves of two artificial plants in the scenes #3, #4, and #6 were considered for the estimation. This way the length of the same leaf was estimated in three different scenarios for a more complete evaluation.

The findings of this analysis show that the length of leaves #1 and #6 that are attached to the bottom and top of the plant was underestimated. For leaf #1 the

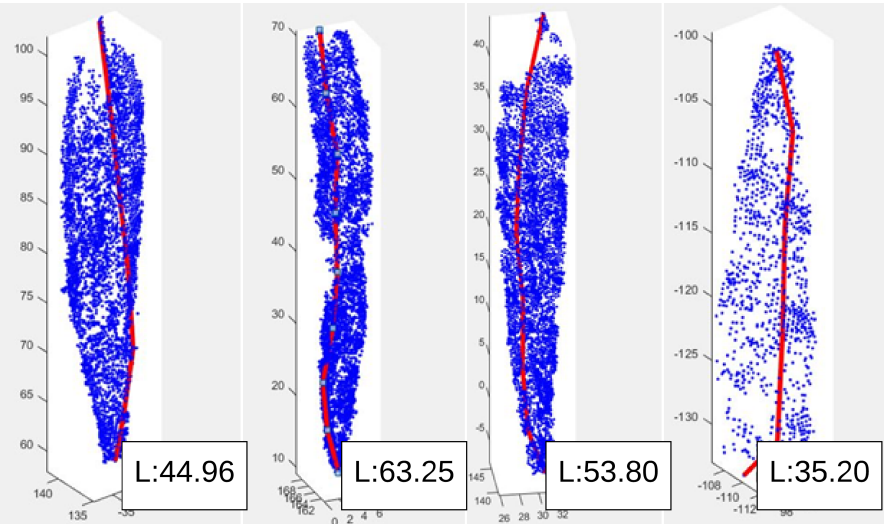


Figure 5.9: The length of four different artificial leaves was automatically measured. The lengths can be seen at the bottom right of each leaf.

Leaf #	1	2	3	4	5	6
Est. μ	35.02	63.27	60.61	42.99	42.49	19.00
Est. σ^2	8.26	3.64	7.40	6.91	3.95	1.83
Real	38.10	58.42	57.15	38.10	36.83	25.40
Error	8.08%	7.67%	5.71%	11.37%	13.32%	25.20%

Table 5.5: This table presents the mean, standard deviation, and the error percentage of the leaf length estimation and leaf length groundtruth of six artificial leaves. This is the value representation of the Fig. 5.10.

partial occlusions from other leaves affected its reconstruction at its base while for #6 the acute angle it shapes with respect to the stem made its separation from the other top leaves (not considered in the experiments) difficult and part of its basis was wrongly segmented. On the other hand, the leaves #2 - #5 were overestimated by almost a constant amount and this is a very interesting finding revealing that the VisualSfM is reconstructing wider leaves. In reality this was hinted by the results of LAI where all the estimated leaf areas were slightly larger than the groundtruth. The accuracy of VisualSfM in creating surfaces such as leaves should be assessed and in the future we should consider different SfM tools.

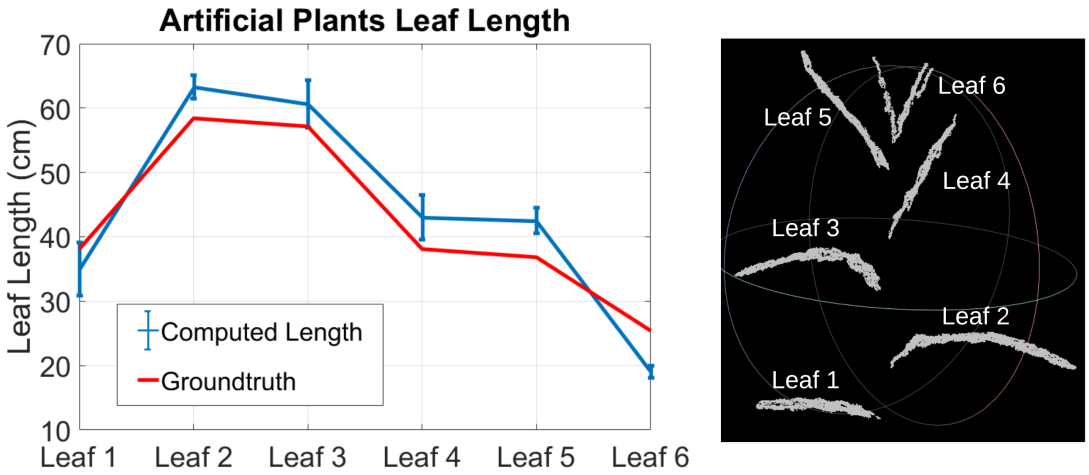


Figure 5.10: The estimated and groundtruthed mean and standard deviation leaf length values of the six major leaves of an artificial corn plant that is seen on the right.

5.5 At the Intersection of Stems and Leaves

Under this category fall a few phenotypes whose estimation depends on the same algorithms since they rely on the 3D points where a leaf and the stem meet. Three such phenotypic characteristics were selected based on their significance for the researchers of corn plants and they are:

1. the leaf count,
2. the leaf angle with respect to the stem, and
3. the inter-nodal distance.

The first component of this estimation is again based on a pipeline of the RAIN algorithm 4.4 where a few iterations segment the plants followed by several more iterations for the stem detection. Each plant is processed separately and, starting from the stem points, a region growing iterative process is looking for leaves. The region between the stem and each leaf is a node of the plant and this information can be used to extract all the aforementioned phenotypes.

The algorithm for the detection of the nodes initially reduces the number of points using a k-means clustering with a large number of clusters for both the stem cloud and

the whole segmented plant. This produces regularly spaced and eroded versions of the initial point clouds and allows for a more clean and efficient solution. The lowest point of the resulting point cloud is set as the root of the plant and a constrained RANSAC is used to determine the best 3D line fit using the following parametric line model:

$$\mathbf{l} = \mathbf{p} + \mathbf{u}t, \tag{5.6}$$

where $\mathbf{l}$ is the line, $\mathbf{p}$ is a point on the line, $\mathbf{u}$ is the direction of the line, and $t \in \mathbb{R}$ is a scalar parameter. The way the constraint RANSAC works is that it requires that the root point is always in the selected random set point.

Once this small number of points on the 3D line have been defined and constrained on the stem, they act as scouts for the detection of leaves. For each one of these points, an increasingly larger radius is searching for neighbors and once a neighbor is found a region growing algorithm expands and captures all the non-stem points of the potential leaf. All the points that belong to the same expanded region are then marked and can not be visited as potential candidates for other stem points.

This method is used to count the number of leaves and pinpoint the base of each leaf and use it to extract the nodes. In turn, the nodes are the points where the angles between leaves and the stem can be extracted. A simple computation of the covariance matrix of the candidate leaf at the points close to the stem reveals the direction of the leaf, similarly to the way the ground plane normal was computed in 4.2. The eigenvector that corresponds to the largest eigenvalue is the direction of the leaf, and the angle can be computed from the inner product of this eigenvector with the direction of the stem as computed in 5.6.

5.5.1 Experimental Results

The results of some real corn plants can be seen in Fig. 5.11 where the height of each node is depicted at the point that is considered the base of each leaf. The different heights can be used to compute the inter-nodal distance, and the nodes of the leaves reveal the leaf-to-stem angle as seen in the visualization of Fig. 5.12.

A very interesting behavior of the leaf-stem angle is captured by Fig. 5.13 that shows

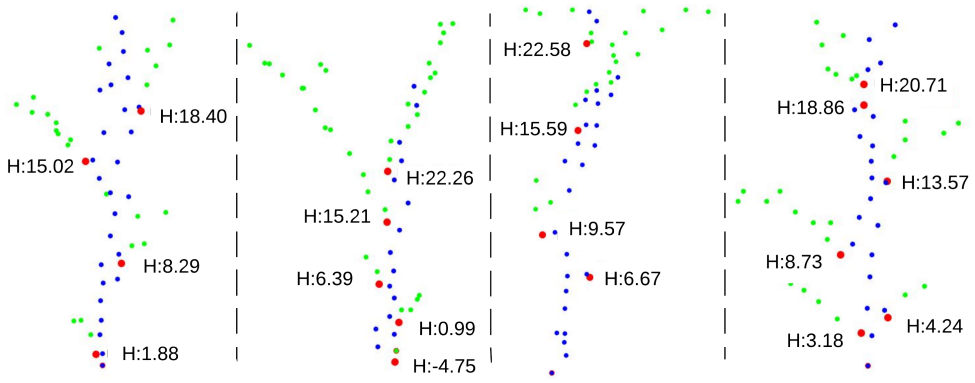


Figure 5.11: Examples of individual plants with their leaves segmented and the height of their nodes denoted. The plants are between "V3" and "V5".

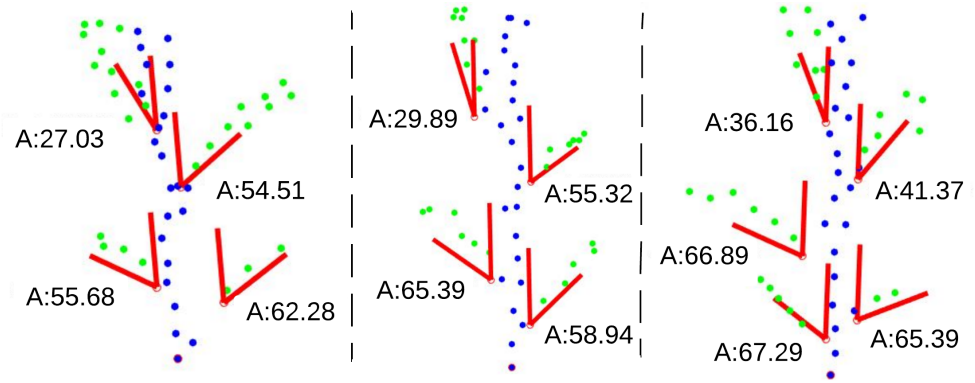


Figure 5.12: Examples of individual plants with automated annotations of the angle between the stem and the leaves. The plants are between "V3" and "V5".

how the leaf angles are distributed with respect to the height. Starting with "V3", the majority of the steep angles are seen towards the top of the plant while, as the plant grows, the angles of the top leaves tend to be less acute. In "V6", there are not many steep angles and majority of the leaves are distributed around the middle of the plant with less acute angles.

Same as the previous phenotypic characteristics, the quality of the 3D reconstruction plays a significant role in the accuracy of the computations. The largest problem comes from the noisy points floating between leaves as they may connect two leaves together during a region growing process. This may result in a wrong number of leaves, inter–nodal distances, and stem-leaf angle estimation.

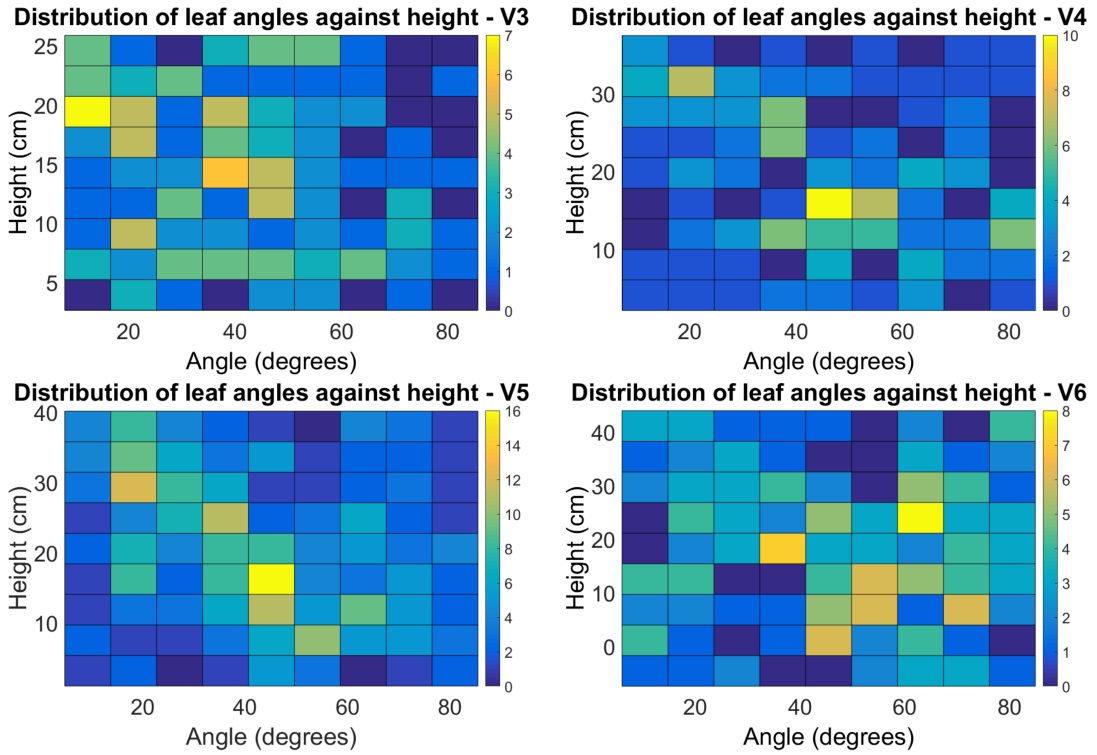


Figure 5.13: The angles (x axis) of several plants going from "V3" to "V6" were plotted against the leaf height (y axis). We can see how the angles are less acute as the plant grows and the leaves that gain more biomass overtime become heavy and tend to lean away from the stem.

5.6 Practical Application

One application is phenotyping maize plants in breeding trials by hybrid developers. Hybrid developers establish hundreds of breeding trials for corn to study the performance and traits of different hybrids as they grow. The objective of these trials is to select the hybrids that have the most desirable traits (e.g. crop yield, resistance to disease, tolerance to drought, etc), and to eliminate from further study those hybrids that fail to meet performance criteria. Through this process the best hybrids are passed on for comparison among each other for additional growing seasons.

Currently, the phenotypic characteristics of the hybrids are assessed using data collected manually from employees that have to collect data from hundreds of acres a day. Therefore, the final decision about which hybrids are most suitable is laborious and somewhat subjective biased by the ability, understanding, and experience of the

employee.

One UAV could collect data from these trials consisting of hundreds of hybrids during one flight, and the processed data could reveal information about phenotypic traits on a per-plant level, allowing the hybrid developers to make more rigorous selection decisions. The proposed methodology introduces consistency in data gathering, and would greatly benefit the corn hybrid breeding and selection industry.

Chapter 6

Conclusion

The proper application of modern CV and ML techniques to precision farming requires a deep understanding of these tools as well as proper knowledge of the important problems in agriculture. It is not common to devise CV algorithms targeting specifically PA applications and is even rarer to apply them successfully to real world scenarios without compromising for the complexity of the environment.

This is where this thesis tried to make a difference and demonstrate that it is possible to put CV and ML expertise to good use and tackle the daunting complexities of a real crop field. The intention was to create a stepping stone for the advancement of the agriculture technology field and particularly the domain of the automated plant pathology assessment through non-destructive and low-cost surveillance.

The research process described throughout this thesis revealed a series of steps taken progressively to achieve a pathological characterization of corn inside its native environment. Starting with the assessment of nutrient deficiencies, we achieved a 90.6% of correct Ndeficiency classification accuracy. Next, realizing the limitations of this method to observe the whole plant we described the 3D reconstruction and point cloud processing pipeline that allowed the detailed segmentation of the main parts of the plants and the extraction of important individual plant phenotypic characteristics. An experimental validation using both artificially made corn plants emulating real world scenarios and real corn plants in different growth stages resulted in estimating the number of plants depicted in a point cloud with 88.1% accuracy, the LAI with 92.48% accuracy, the height with 89.2% accuracy, the leaf length with 74.8% accuracy, and

the location and the angles of leaves with respect to the stem. The last two variables together showed the trend of the angles to change with respect to the leaf position on the stem as the crops grow. Having stated this, it is important to realize that this thesis is not an exhaustive exploration of the solutions for the proposed problems.

Future research endeavors need to overcome the limitations stated throughout the chapters and enhance the robustness of the algorithms. The detection of nutrient deficiencies needs to be applied to a broader selection of deficiencies and crops and the optimal altitude to coverage ration needs to be determined with extensive imagery testing. Further, it is vital to develop a 3D reconstruction pipeline directed towards agricultural needs; something that takes into account the motion of plants due to wind and fuses the reconstruction with the fitting of computerized models of a corn plant. The incorporation of simulated corn plant models could significantly increase the accuracy of the phenotype extraction and can reduce the noise from the 3D reconstruction.

PA is a vast domain with a plethora of applications and unsolved problems and has not yet received proper attention from the CV and robotics community. It is an open domain full of opportunities to devise better and more elegant solutions using technology that is already available for other applications such as virtual reality or autonomous driving. It is our hope for this thesis to be viewed as a medium for the development of the next generation agriculture technology, one that treats the needs of each individual plant and acts based on frequent and timely collected information.

Acknowledgment of Funding Sources

This work was supported by the Minnesota Corn Research and Promotion Council, the Minnesota Corn Growers Association, and the National Science Foundation with the following grants: #IIP-0934327, #IIP-1032018, #IIS-1017344, #CNS-1061489, #CNS-1138020, #IIP-1127938, #IIP-1237259, #IIP-1332133, #IIS-1427014, #IIP-1432957, #CNS-1439728, and #CNS-1531330

References

[1] K. Skunes and C. Novak. World of corn. *National Corn Growers Association (NCGA)*, 2018.

[2] J. M. MacDonald, P. Korb, and R. A. Hoppe. Farm size and the organization of u.s. crop farming. *Economic Research Report*, (152), 2013.

[3] U. Schlüter, M.n Mascher, C. Colmsee, U. Scholz, A. Bräutigam, H. Fahnenstich, and U. Sonnewald. Maize source leaf adaptation to nitrogen deficiency affects not only nitrogen and carbon metabolism but also control of phosphate homeostasis. *Plant Physiology*, 160(3):1384–1406, 2012.

[4] J. J Hanway et al. How a corn plant develops. *How a corn plant develops*, 1966.

[5] N. Viswanadham, S. Chidananda, Y. Narahari, and P. Dayama. Mandi electronic exchange: Orchestrating indian agricultural markets for maximizing social welfare. In *IEEE International Conference on Automation Science and Engineering (CASE), 2012*, pages 992–997. IEEE, 2012.

[6] Y. Lin, M. Kang, and J. Hua. Fitting a functional structural plant model based on global sensitivity analysis. In *IEEE International Conference on Automation Science and Engineering (CASE), 2012*, pages 790–795. IEEE, 2012.

[7] J. Das, G. Cross, C. Qu, A. Makineni, P. Tokekar, Y. Mulgaonkar, and V. Kumar. Devices, systems, and methods for automated monitoring enabling precision agriculture. In *IEEE International Conference on Automation Science and Engineering (CASE), 2015*, pages 462–469. IEEE, 2015.

[8] D. V. Gealy, S. McKinley, M. Guo, L. Miller, S. Vougioukas, J. Viers, S. Carpin, and K. Goldberg. Date: A handheld co-robotic device for automated tuning of emitters to enable precision irrigation. In *IEEE International Conference on Automation Science and Engineering (CASE), 2016*, pages 922–927. IEEE, 2016.

[9] D. J. Mulla. Twenty five years of remote sensing in precision agriculture: Key advances and remaining knowledge gaps. *Biosystems Engineering*, 114(4):358–371, 2013. Special issue: Sensing Technologies for Sustainable Agriculture.

[10] J. G. A. Barbedo. Digital image processing techniques for detecting, quantifying and classifying plant diseases. *SpringerPlus*, 2(1):660, 2013.

[11] J. G. A. Barbedo. A review on the main challenges in automatic plant disease identification based on visible range images. *Biosystems Engineering*, 144:52 – 60, 2016.

[12] J. Wang, J. He, Y. Han, C. Ouyang, and D. Li. An adaptive thresholding algorithm of field leaf image. *Computers and Electronics in Agriculture*, 96:23–39, 2013.

[13] W. Guo, U. K. Rage, and S. Ninomiya. Illumination invariant segmentation of vegetation for time series wheat images based on decision tree model. *Computers and Electronics in Agriculture*, 96:58–66, 2013.

[14] X. D. Bai, Z. G. Cao, Y. Wang, Z. H. Yu, X. F. Zhang, and C. N. Li. Crop segmentation from images by morphology modeling in the cie L* a* b* color space. *Computers and Electronics in Agriculture*, 99:21–34, 2013.

[15] A. K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.

[16] L. F. Tian and D. C. Slaughter. Environmentally adaptive segmentation algorithm for outdoor image segmentation. *Computers and Electronics in Agriculture*, 21(3):153–168, 1998.

[17] A. Camargo and J. S. Smith. An image-processing based algorithm to automatically identify plant disease visual symptoms. *Biosystems Engineering*, 102(1):9–21, 2009.

[18] G. Ruiz-Ruiz, J. Gómez-Gil, and L. M. Navas-Gracia. Testing different color spaces based on hue for the environmentally adaptive segmentation algorithm (EASA). *Computers and Electronics in Agriculture*, 68(1):88–96, 2009.

[19] M. Guijarro, G. Pajares, I. Riomoros, P. J. Herrera, X. P. Burgos-Artizzu, and A. Ribeiro. Automatic segmentation of relevant textures in agricultural images. *Computers and Electronics in Agriculture*, 75(1):75–83, 2011.

[20] J. L. Hernández-Hernández, G. García-Mateos, J. M. González-Esquiva, D. Escarabajal-Henarejos, A. Ruiz-Canales, and J. M. Molina-Martínez. Optimal color space selection method for plant/soil segmentation in agriculture. *Computers and Electronics in Agriculture*, 122:124–132, 2016.

[21] T. Gevers and A. W. M. Smeulders. Color-based object recognition. *Pattern recognition*, 32(3):453–464, 1999.

[22] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

[23] H. Lu, Z. Cao, Y. Xiao, Y. Li, and Y. Zhu. Region-based colour modelling for joint crop and maize tassel segmentation. *Biosystems Engineering*, 147:139–150, 2016.

[24] M. Ye, Z. Cao, Z. Yu, and X. Bai. Crop feature extraction from images with probabilistic superpixel markov random field. *Computers and Electronics in Agriculture*, 114:247–260, 2015.

[25] M. J. Afridi, X. Liu, and J. M. McGrath. An automated system for plant-level disease rating in real fields. In *22nd International Conference on Pattern Recognition (ICPR)*, pages 148–153. IEEE, 2014.

[26] A. Camargo and J. S. Smith. Image pattern classification for the identification of disease causing agents in plants. *Computers and Electronics in Agriculture*, 66(2):121 – 125, 2009.

[27] L. M. Romualdo, P. H. C. Luz, F. F. S. Devechio, M. A. Marin, A. M. G. Ziga, O. M. Bruno, and V. R. Herling. Use of artificial vision techniques for diagnostic of nitrogen nutritional status in maize plants. *Computers and Electronics in Agriculture*, 104:63 – 70, 2014.

[28] M. Zhang and Q. Meng. Automatic citrus canker detection from leaf images captured in field. *Pattern Recognition Letters*, 32(15):2036–2046, 2011.

[29] R. D. L. Pires, D. N. Gonçalves, J. P. M. Oruê, W. E. S. Kanashiro, J. F. Rodrigues, B. B. Machado, and W. N. Gonçalves. Local descriptors for soybean disease recognition. *Computers and Electronics in Agriculture*, 125:48–55, 2016.

[30] D.W. Stewart and L.M. Dwyer. Mathematical characterization of maize canopies. *Agricultural and Forest Meteorology*, 66(3-4):247–265, nov 1993.

[31] C. Fournier and B. Andrieu. A 3d architectural and process-based model of maize development. *Annals of Botany*, 81(2):233–250, 1998.

[32] H. Sinoquet, S. Thanisawanyangkura, H. Mabrouk, and P. Kasemsap. Characterization of the light environment in canopies using 3d digitising and image processing. *Annals of Botany*, 82(2):203–212, 1998.

[33] I. Kalisperakis, Ch. Stentoumis, L. Grammatikopoulos, and K. Karantzalos. Leaf area index estimation in vineyards from uav hyperspectral data, 2d image mosaics and 3d canopy surface models. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(1):299, 2015.

[34] J. Guo and L. Xu. Automatic segmentation for plant leaves via multiview stereo reconstruction. *Mathematical Problems in Engineering*, 2017, 2017.

[35] A. Paproki, X. Sirault, S. Berry, R. Furbank, and J. Fripp. A novel mesh processing based technique for 3D plant analysis. *BMC Plant Biology*, 12, 2012.

[36] H. Wang, W. Zhang, G. Zhou, G. Yan, and N. Clinton. Image-based 3d corn reconstruction for retrieval of geometrical structural parameters. *International Journal of Remote Sensing*, 30(20):5505–5513, 2009.

[37] Renato P. de Moraes F. and W. F. Krajewski. Three-dimensional digital model of a maize plant. *Agricultural and Forest Meteorology*, 150(3):478–488, 2010.

[38] T. Santos and A. De Oliveira. Image-based 3d digitizing for plant architecture analysis and phenotyping. In *Embrapa Informática Agropecuária-Artigo em anais de congresso (ALICE)*. In: CONFERENCE ON GRAPHICS, PATTERNS AND IMAGES, 25., 2012, Ouro Preto. Workshop on industry applications.[Sl]: Conference Publishing Series, 2012., 2012.

[39] N. Ivanov, P. Boissard, M. Chapron, and B. Andrieu. Computer stereo plotting for 3-d reconstruction of a maize canopy. *Agricultural and Forest Meteorology*, 75(1-3):85–102, 1995.

[40] P. Sodhi, S. Vijayarangan, and D. Wettergreen. In-field segmentation and identification of plant structures using 3d imaging. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017.

[41] J. Jin and L. Tang. Corn plant sensing using real-time stereo vision. *Journal of Field Robotics*, 2009.

[42] S. Jay, G. Rabatel, X. Hadoux, D. Moura, and N. Gorretta. In-field crop row phenotyping from 3D modeling performed using Structure from Motion. *Computers and Electronics in Agriculture*, 110:70–77, 2015.

[43] B. Biskup, H. Scharr, U. Schurr, and U. Rascher. A stereo imaging system for measuring structural parameters of plant canopies. *Plant Cell Environment*, 30:12991308, Jun 2007.

[44] J. Dong, L. Carlone, G. C. Rains, T. Coolong, and F. Dellaert. 4D mapping of fields using autonomous ground and aerial vehicles. *ASABE and CSBE/SCGAB Annual International Meeting*, 2014.

[45] L. Carlone, J. Dong, S. Fenu, G. C. Rains, and F. Dellaert. Towards 4D crop analysis in precision agriculture: Estimating plant height and crown radius over time via expectation-maximization. *Workshop on Robotics in Agriculture, IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[46] C. Qu, G. Cross, A. Makineni, Y. Mulgaonkar, J. Das, P. Tokekar, and V. Kumar. A versatile imaging system for automated monitoring enabling precision agriculture. *Workshop on Robotics in Agriculture, IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[47] K. Kjaer and C. O. Ottosen. 3d laser triangulation for plant phenotyping in challenging environments. *Sensors*, 15(6):1353313547, Jun 2015.

[48] M. L. España, F. Baret, F. Aries, M. Chelle, B. Andrieu, and L. Prévot. Modeling maize canopy 3D architecture: Application to reflectance simulation. *Ecological Modelling*, 122(1):25–43, 1999.

[49] M. España, F. Baret, F. Aries, B. Andrieu, and M. Chelle. Radiative transfer sensitivity to the accuracy of canopy structure description. the case of a maize canopy. *Agronomie*, 19(3-4):241–254, 1999.

[50] C. Fournier, C. Pradal, K. Meng Zhen, D. Yves, and G. Yan. A Plastic, Dynamic and Reducible 3D Geometric Model for Simulating Gramineous Leaves. *International Symposium on Plant Growth Modeling, Simulation*, pages 125–132, 2012.

[51] C. Pradal, S. Dufour-Kowalski, F. Boudon, C. Fournier, and C. Godin. Openalea: a visual programming and component-based software platform for plant modelling. *Functional plant biology*, 35(10):751–760, 2008.

[52] F. Li, B. Mistele, Y. Hu, X. Chen, and U. Schmidhalter. Reflectance estimation of canopy nitrogen content in winter wheat using optimised hyperspectral spectral indices and partial least squares regression. *European Journal of Agronomy*, 52:198–209, 2014.

[53] W. Kazmi, S. Foix, G. Alenyà, and H. J. Andersen. Indoor and outdoor depth imaging of leaves with time-of-flight and stereo vision sensors: Analysis and comparison. *ISPRS Journal of Photogrammetry and Remote Sensing*, 88:128–146, 2014.

[54] K. Omasa, F. Hosoi, and A. Konishi. 3D lidar imaging for detecting and understanding plant responses and canopy structure. *Journal of Experimental Botany*, 58(4):881–898, nov 2006.

[55] R. Klose, J. Penlington, and A. Ruckelshausen. Usability study of 3d time-of-flight cameras for automatic plant phenotyping. *Image Analysis for Agricultural Products and Processes*, page 93105, 2009.

[56] G. Alenya, B. Dellen, and C. Torras. 3D modelling of leaves from color and ToF data for robotized plant measuring. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3408–3414, May 2011.

[57] C. Bellasio, J. Olejníčková, R. Tesa, D. Šebela, and L. Nedbal. Computer Reconstruction of Plant Growth and Chlorophyll Fluorescence Emission in Three Spatial Dimensions. *Sensors*, 12(12):1052–1071, jan 2012.

[58] S. L. Osborne, J. S. Schepers, D. D. Francis, and M. R. Schlemmer. Detection of phosphorus and nitrogen deficiencies in corn using spectral radiance measurements. *Agronomy Journal*, 94(6):1215–1221, 2002.

[59] A. Bechar and C. Vigneault. Agricultural robots for field operations: Concepts and components. *Biosystems Engineering*, 149:94–111, 2016.

[60] M. Quemada, J. L. Gabriel, and P. Zarco-Tejada. Airborne hyperspectral images and ground-level optical sensors as assessment tools for maize nitrogen fertilization. *Remote Sensing*, 6(4):2940–2962, 2014.

[61] T. M. Blackmer, J. S. Schepers, G. E. Varvel, and E. A. Walter-Shea. Nitrogen deficiency detection using reflected shortwave radiation from irrigated corn canopies. *Agronomy Journal*, 88(1):1–5, 1996.

[62] Mississippi river/gulf of mexico watershed nutrient task force. *Report to Congress*, 2015.

[63] R. Alley. Algorithm theoretical basis document for decorrelation stretch. JPL AST06, 1999.

[64] S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[65] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893, 2005.

[66] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague, 2004.

[67] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 689–696. ACM, 2009.

[68] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice. *International Journal of Computer Vision*, 105(3):222–245, 2013.

[69] C. Wu et al. VisualSFM: A visual structure from motion system. 2011.

[70] M. Y. Yang and W. Förstner. Plane detection in point cloud data. In *Proceedings of the 2nd International Conference on Machine Control Guidance, Bonn*, volume 1, pages 95–104, 2010.

[71] M. Ester, H. P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. *KDD*, 96:226–231, 1996.

[72] R. B. Rusu. *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. PhD thesis, Computer Science department, Technische Universitaet Muenchen, Germany, October 2009.

[73] J. Cao, A. Tagliasacchi, M. Olson, H. Zhang, and Z. Su. Point cloud skeletons via Laplacian based contraction. In *Shape Modeling International Conference (SMI), 2010*, pages 187–197. IEEE, 2010.

[74] E. G. Montgomery. Correlation studies in corn. In *24th Annual Report, Nebraska, Mo, USA*. Agricultural Experiment Station, 1911.

[75] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(8):1362–1376, Aug 2010.

[76] D. J. Watson. Comparative physiological studies on the growth of field crops: I. variation in net assimilation rate and leaf area between species and varieties and within and between years. *Annals of Botany*, 11:41–76, 1947.

[77] I. Jonckheere, S. Fleck, K. Nackaerts, B. Muys, P. Coppin, M. Weiss, and F. Baret. Review of methods for in situ leaf area index determination: Part i. theories, sensors and hemispherical photography. *Agricultural and forest meteorology*, 121(1):19–35, 2004.

[78] J. M. Chen, P. S. Plummer, M. Rich, S. T. Gower, and J. M. Norman. Leaf area index measurements. *Journal of Geophysical Research*, 102(D24):29–429, 1997.

[79] N. J. J. Bréda. Ground-based measurements of leaf area index: a review of methods, instruments and current controversies. *Journal of experimental botany*, 54(392):2403–2417, 2003.

[80] W. W. Wilhelm, K. Ruwe, and M. R. Schlemmer. Comparison of three leaf area index meters in a corn canopy. *Crop Science*, 40(4):1179–1183, 2000.

[81] R. Hu, G. Yan, X. Mu, and J. Luo. Indirect measurement of leaf area index on the basis of path length distribution. *Remote Sensing of Environment*, 155:239 – 247, 2014.

[82] G. Zheng and L. M. Moskal. Retrieving leaf area index (lai) using remote sensing: Theories, methods and sensors. *Sensors*, 9(4):2719–2745, 2009.

[83] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proceedings of the 4th Eurographics Symposium on Geometry Processing*, volume 7, 2006.

[84] T. Kohonen. Self-organizing maps. *Series in Information Sciences*, 30, 1997.

Chapter 7

Glossary and Acronyms

Care has been taken in this thesis to minimize the use of jargon and acronyms, but this cannot always be achieved. This appendix defines jargon terms in a glossary, and contains a table of acronyms and their meaning.

7.1 Glossary

- **Precision Agriculture** (**PA**) – PA is a research area concerned with the yield maximization of farm fields by applying the right farm inputs at the right place and time.
- **Remote Sensing** (**RS**) – RS is the science of obtaining information about objects or areas from a distance.
- **3D Reconstruction** – The process of creating a 3D model of an object by observing its lower dimensional (2D or 1D).
- **Structure from Motion** (**SfM**) – SfM is a concept in Computer Vision for the creation of 3D reconstruction models from a series of 2D images utilizing the rotation and translation of the camera sensor with respect to the depicted object.
- **Segmentation** – Segmentation is a broad term used to describe the separation of data into homogeneous groups. The characteristics of the homogeneity are defined

by the user. Example: For image segmentation pixels belong to the same group if they have similar color value or x-y coordinates.

- **Classification** – The process of assigning a label to an object by evaluating a model which is a product of training. The training of the model involves a set of objects along with their labels. Example: Given a set of pixels labeled as green or red, a trained model is produced that is able to classify a newly arrived pixel as green or red based on its color value.

- **Clustering** – The process of assigning a label to an object by evaluating a set of rules which define a concept of similarity. Clustering does not involve a training step. Example: Given a set of unlabeled pixels, group them together in an unknown number of groups based on their similarity in the color space and their position in the image (This is a high level description of a super-pixel generation algorithm such as SLIC[22]).

7.2 Acronyms

Table 7.1: Acronyms

Acronym	Meaning
PA	Precision Agriculture
RS	Remote Sensing
CV	Computer Vision
ML	Machine Learning
N	Nitrogen
3D	Three Dimensional
2D	Two Dimensional
SfM	Structure from Motion
SLIC	
UAV	Unmanned Aerial Vehicle
LiDAR	Light Detection And Ranging

Continued on next page

Table 7.1 – Continued from previous page

Acronym	Meaning
USDA	United States Department of Agriculture
NIR	Near Infra-Red
ROI	Region of Interest
SVM	Support Vector Machines
VPI	Vectorized Pixel Intensities
NCH	Normalized Color Histograms
HOG	Histograms of Oriented Gradients
BVW	Bags of Visual Words
SC	Sparse Codes
FV	Fischer Vectors
SIFT	Scale Invariant Feature Transform
GMM	Gaussian Mixture Models
RGB	Red Green Blue
RAIN	Randomly Intercepted Nodes
SKF	Skeleton Kalman Filter